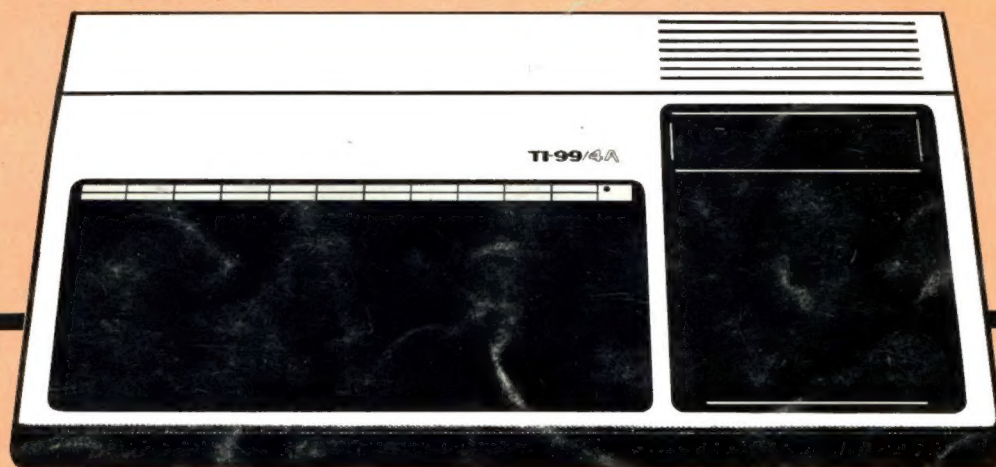


99 MAGAZINE

CASSETTE
D'ACCOMPAGNEMENT
DISPONIBLE

Crayon optique
Mission Spéciale
Comptes familiaux
Bit-Map et Mini-mémoire



Numéro 8 - Trimestriel -

Mars 1985 - 40 F

Sommaires des numéros 1 à 7 de "99 Magazine"

Numéro 1

Musik Maker à l'essai
Pourquoi acheter un TI-99/4A
A vous de programmer
Déballage du TI-99
Les avantages du Basic Etendu
Initiation au Basic
Jeu du Pendu
Fabriquez votre joystick
Appelez moi KEY
Programmes d'aide à la décision
Jeu de l'Alphabet
Le club d'utilisateur TI-Education
Introduction au LOGO
Les tours de Hanoi
Récupérez les programmes Apple
Le troisième menuet de Back
Le dernier robot
Cherchez l'ancêtre
Gestion Personnelle de Fichiers
Quand le TI devient un harmonium
Fichiers en Basic

Numéro 2

Micro-monde LOGO : les lutins
La bataille navale
Le crocodile savant
Jeu du Simon
Jeu des Carrés
L'ordinateur de l'avenir
TI-Writer à l'essai
Canon de Purcell
Chicken Helper
Sauvegarde des fichiers et programmes
Le Mastermind
Othello
Astéroïdes
Kermit la grenouille
Le Blorhythme

Numéro 3

Un programme de calcul
Annonce de Texas Instruments
Othello à l'essai
Apprendre le Basic en Programmant (1)
Introduction au Pascal
Puissance 4
Les modes d'adressage en assembleur
Géographie : la carte de France
Créez votre touche RESET
Puzzle
Awari
Le sapin de Noël
PARSEQ à l'essai
Logo, seulement un langage pour enfants ?
Le Mastermind en assembleur
Du nouveau sur les cassettes
Le Magnétophone TI

Numéro 4

SOS Hélicoptère
Utilisation de la Mini-mémoire
Communiqué de Texas Instruments France
Le serveur de restaurant
"Echecs" à l'essai
Combat de chars
Réalisez votre cordon de magnétophone
Apprendre le Basic en programmant (2)
Jeu de mémorisation
Jack-pot
Les nombres en Logo
Graphisme en haute résolution
Sous-programmes en assembleurs

Horloge
Un détecteur de lumière pour le TI-99
Le club M. T. I.
Le système P-UCSD
Les lutins du Basic Etendu
Le solitaire
Claustrophobie
"DISPLAY" & "ACCEPT" avec le module "PRK"
Une erreur dans le manuel de la Mini-mémoire
Réparez votre assembleur ligne par ligne

Numéro 5

Conversions
Systèmes linéaires
Structure des disques du P-système
Poèmes et graphismes
Un désassembleur en assembleur
Les tours de Hanoi
Trouvez le bon mot
Le jeu de la vie
Programmathèque
Les opérateurs logiques
Apprendre le Basic en programmant (3)
Bibliographie
Warangal
Graphismes en haute résolution
Le trésor de l'Armada
Ecriture littérale d'un nombre
Copie d'écran sur imprimante Epson
Description du port d'entrées/sorties
Translet
Le club Ordinabis
Le club Ticazur

Numéro 6

Des chiffres et des lettres
A l'heure du Pascal
Kong
Mini-mémoire moins "mini"
Edipage
Caractères ASCII dans la page graphique
Musik
Copie d'écran sur Seikosha GP-100
Nouvelles fonctions avec la Mini-mémoire
Le Basic à la loupe
Star trek
Ecriture directe dans les registres VDP
My tailor was rich
Le code de la route
Un double face sur votre TI

Numéro 7

La défasse
Le tableur TI-CALC
Sauvegarde d'images graphiques
et copie d'écran
Copie d'écran sur Seikosha GP-250
La compilation séparée et les bibliothèques
Trouvez votre chemin dans
les jeux d'aventures TI
Srand
Galaxia
LOGO et la récursivité
Dessinez avec le Basic étendu
Accès au mode BIT-MAP en Basic TI
Music Box Dancer
Des nouvelles du club Ticazur
Routines graphiques pour le Basic étendu

* BTI : Basic TI. BE : Basic Etendu. A : Assembleur.
P : Pascal. L : Logo.

Tous les numéros sont disponibles.

SOMMAIRE

— 5 —

Editorial

Hervé Thiriez

— 6 —

**Bit-Map et
Mini-mémoire**

Gérard Baroni
Jean Marin

— 10 —

**Le train sifflera
trois fois...**

Denise Amrouche
Gilbert Arribet
Jean Sahel

— 15 —

**Comptes
familliaux**

Jean-philippe
Guillemant

99 MAGAZINE

— 23 —

Mission spéciale

Edouard Krok

— 29 —

Crayon optique

Bernard Bezuel

— 54 —

**Le traitement
des fichiers
en Pascal**

Gérard
Santraille

— 60 —

**Editeur de
caractères**

Julien Thomas

— 61 —

Kaleïdoscope

Marianne
Sutz

— 62 —

Opérations

Albert Rigal

— 65 —

**Petites
annonces...**

gratuites

— 67 —

**Courrier
des
lecteurs**

Alexandre
Duback

Editions MEV - 64/70, rue des Chantiers - 78000 Versailles

Cassette 99A

Basic TI

- Electronique
- Mineur

Basic Etendu

- Prêt

Cassette 99B

Basic TI

- Yathzee
- Division

Basic Etendu

- Chardef

Cassette 99C

Basic TI

- Division
- Course

Basic Etendu

- Dames
- Nim
- Isola
- Schmoo

Cassette 99D

Basic TI

- Caractérologie
- Car driver
- Cannibales

Basic Etendu

- Conjugaisons
- Poker

Cassette 99E

Basic TI

- Mic-Math
- Course de chevaux
- Jeu de poursuite
- Jeu de dés

Basic Etendu

- Guerre atomique
- Course en ligne droite

Cassette

Mastermind

Assembleur. Fonctionne avec la Mini-mémoire seule.

Disquette

Mastermind

Assembleur. Fonctionne à partir du module "Basic Etendu".

Disquette Routines Graphiques

Assembleur. Fonctionne avec le module "Editeur/Assembleur" ou le module "Mini-mémoire".

Disquette

Désassembleur

Assembleur. Fonctionne avec les modules "Editeur/Assembleur", "Mini-Mémoire" et "Basic Etendu".

Disquette Galaxia

Assembleur. Fonctionne avec les modules "Editeur/Assembleur" ou "Mini-mémoire".

Disquette

Crayon optique

Assembleur. Fonctionne avec les modules "Editeur/Assembleur" et "Mini-mémoire".

99MAGAZINE

BON DE COMMANDE

- | | |
|---|----------|
| <input type="checkbox"/> 99A : Mineur, Prêt, Electronique | 50 F TTC |
| <input type="checkbox"/> 99B : Yahtzee, Chardef, Division, Régression linéaire | 50 F TTC |
| <input type="checkbox"/> 99C : Dames, Nim, Division 2, Isola, Schmoo, Robots | 50 F TTC |
| <input type="checkbox"/> 99D : Conjugaisons, caractérologie, Car driver, Poker, cannibales | 50 F TTC |
| <input type="checkbox"/> 99E : Mic-Math, Course de chevaux, Poursuite, Jeu de dés, Guerre atomique, Course en ligne droite | 50 F TTC |
| <input type="checkbox"/> Cassette "Mastermind" | 50 F TTC |
| <input type="checkbox"/> Disquette "Mastermind" (cf. 99 Magazine N° 3) | 55 F TTC |
| <input type="checkbox"/> Disquette "Routines graphiques" (cf. 99 Magazine N° 4, 5, 6 et 7) | 55 F TTC |
| <input type="checkbox"/> Disquette "Désassembleur" (cf. 99 Magazine N° 5) | 55 F TTC |
| <input type="checkbox"/> Disquette "Galaxia" (cf. 99 Magazine N° 7) | 55 F TTC |
| <input type="checkbox"/> Disquette "Crayon optique" (cf. 99 Magazine N° 8) | 55 F TTC |

Nom

Adresse

Ces tarifs comprennent l'envoi postal en France Métropolitaine et CEE (voie aérienne exceptée)

Envoi par avion : + 10 F

Envoyez ce bon de commande et votre règlement à :

Editions MEV - 64-70, rue des Chantiers - 78000 Versailles

99 Magazine est-il toujours publié ?... c'est une question que l'on nous pose très souvent au téléphone, ou lors de salons tel le récent Micro-Expo. Ceci est certainement dû au fait que de nombreuses personnes pensent que nous avons un lien de parenté avec Texas Instruments, ce qui les laisse croire que "99 Magazine" a cessé d'exister au moment de l'arrêt de production du TI-99/4A. Quoi qu'il en soit, notre revue est bien présente et vous êtes toujours aussi nombreux à nous être fidèles, ce qui prouve qu'il est possible de publier une revue spécialisée bien après la fin de commercialisation d'un matériel. Passons maintenant au contenu de ce numéro.

Gérard Baroni et Jean Marin, nos deux amis du Club Ticazur, travaillent toujours avec acharnement sur la Mini-mémoire et nous confient le résultat de leurs investigations. Bernard Bezuel, quant à lui, nous propose un programme exceptionnel puisqu'il s'agit, à notre connaissance, du premier programme graphique pour TI-99 fonctionnant avec un crayon optique. Vous trouverez dans cet article les instructions et schémas permettant la réalisation du crayon. Les sources comportent de très nombreuses remarques fort intéressantes et nous vous conseillons vivement d'étudier la routine "REMPLISSAGE", remarquable...

Denise Amrouche, Gilbert Arribet et Jean Sahal ont regroupé leurs efforts afin de vous communiquer un programme où les possibilités sonores et graphiques du Basic Etendu sont pleinement exploitées. Vous connaissez tous la qualité des programmes écrits par Edouard Krok; il récidive avec le programme "Mission spéciale", dans lequel il vous faudra affronter l'abominable Krapul et ses acolytes.

Editorial

Vous étiez nombreux à nous demander un programme de gestion de comptes familiaux; c'est désormais chose faite. Celui-ci a été écrit par Jean-Philippe Guillemant et fonctionne avec un lecteur de disquette et une extension 32Ko. Encore une fois, nous vous conseillons l'analyse du programme, écrit de façon remarquable. Il faut signaler que, sur la cassette d'accompagnement, le programme est scindé en deux parties qu'il vous faudra relier avec la classique instruction MERGE.

Gérard Santraille poursuit sa série d'initiations au Pascal (on ne s'en lasse pas...) fort appréciée pour sa clarté et sa justesse par les amateurs de ce langage de haut niveau. "Opérations" est un programme à caractère éducatif, écrit en Basic TI par Albert Rigal.

Julien Thomas nous propose un programme permettant la définition de caractères ou lutins et, pour conclure, Marianne Sutz vous communique un très court programme pour démontrer que le TI autorise des effets graphiques intéressants avec seulement quelques lignes de Basic.

Hervé Thiriez

Rédacteur en chef - Directeur de la publication : Hervé Thiriez. **Comité de rédaction :** Jean-Luc Bazanegue, Gérard Santraille. **Dessins :** Laurent Bidot. **Ont collaboré à ce numéro :** Denise Amrouche, Gilbert Arribet, Gérard Baroni, Jean-Luc Bazanegue, Bernard Bezuel, Alexandre Duback, Jean-Philippe Guillemant, Edouard Krok, Jean Marin, Albert Rigal, Jean Sahal, Gérard Santraille, Marianne Sutz, Julien Thomas.

Editions MEV - 64/70, rue des Chantiers - 78000 Versailles.

Publicité : consultez les Editions MEV.

Impression : S.N.E. Repro-Versailles - 21, rue Danièle Casanova - 78210 St Cyr l'école - Tél. : 460.38.80.

Bit-Map et Mini-mémoire

Gérard Baroni et Jean Marin

N.D.L.R. : les programmes proposés avec cet article ne fonctionnent pas si un contrôleur de disquettes est raccordé au TI.

Dans ce troisième article, nous rappelons que notre but est de rendre compatible le mode Bit-Map - dont on connaît la puissance graphique - et le Basic dont on apprécie la souplesse d'édition.

La configuration minimale exigée pour mettre en œuvre les programmes décrits dans cet article est : console, mini-mémoire et lecteur de K7.

Après l'écriture directe dans les registres VDP et l'accès au mode Bit-Map en Basic (14 Ko pour l'image et 2 Ko pour le Basic) et comme nous vous l'annonçons dans le dernier article, nous mettons à votre disposition les listings de routines en assembleur développées dans le cadre des activités du Club TICAZUR.

Ces routines, à implanter dans votre mini-mémoire, seront appelées du Basic par des CALL LINK ("BITMAP"), CALL LINK ("COLOR"), CALL LINK ("POINT",Y,X), CALL LINK ("DROITE",YD,XD,YA,XA).

Vous avez déjà compris, avec ces noms évocateurs, que vous allez pouvoir travailler sur des graphismes haute résolution avec toute la puissance et la vitesse d'exécution du langage machine et, toujours selon notre principe, avec la facilité d'emploi et d'édition du Basic. Désormais, les 2 Ko disponibles en Basic sont totalement

libérés et utilisables pour des calculs, saisies, etc...

Les programmes dont les listings sont joints démontrent qu'en quelques lignes Basic vous pouvez arriver, rapidement, à des applications graphiques élaborées.

Quelques remarques sur ces programmes

LISSAJOUS

Comparez sa vitesse d'exécution et sa place en mémoire avec le programme Basic que nous vous avons proposé dans le numéro 7 de 99 Magazine.

KALEIDOSCOPE

Il vous est parfaitement possible de modifier ce programme et d'ajouter ou retrancher des miroirs. Vous ralentirez ou accélérerez l'exécution selon votre option.

POLYGONE

Exemples :

- un angle de 10° donne un cercle;
- un angle de 150° donne une étoile à 12 pointes;
- un angle de 85° donne une couronne dont vous trouverez, joint à cet article, une copie d'écran sortie sur imprimante Seikosha GP100A.

LINES

Voici une version Basic du programme de démonstration "LINES" fourni par Texas Instruments avec la Mini-mémoire.

Nous ne pouvons pas, dans un article tel que celui-ci, commenter les routines et les programmes fournis comme nous le faisons à notre club dans notre compte-rendu hebdomadaire (nous en sommes actuellement au numéro 23). Mais, si vous le souhaitez bien sûr, nous avons le plaisir de vous annoncer que d'autres sous-programmes en assembleur pour votre Mini-mémoire, baptisés CALL LINK ("LI"), ("SI"), ("CSION"), ("CSIOFF") et autres ("TEXTE") ou ("DIM"), seront bientôt à votre disposition.

Alors, lisez 99 Magazine évidemment, dans lequel nous espérons dans un avenir proche vous donner notre solution à quelques problèmes. Par exemple, maintenant que nous sommes capables de créer des dessins, il serait souhaitable de pouvoir les sauvegarder; or, les routines Texas (disquettes ou cassettes) ne conviennent pas pour des raisons telles que :

- un buffer en RAM VDP ne laisse que très peu de place;
- affichage des messages non compatible avec le mode Bit-Map.

Notre cher vieux TI-99/4A, malgré son âge, a encore beaucoup à nous offrir et, avec sa Mini-mémoire chargée avec des sous-programmes en assembleur, il sera capable encore longtemps de rivaliser avec les meilleurs.



LISSAJOUS

```
10 REM LISSAJOUS
100 CALL LOAD(-31890,56,1)
110 CALL LINK("BITMAP")
120 YD=186
130 XD=128
140 FOR T=1 TO 360 STEP 4
150 C=16*INT(2+14*RND)
160 CALL LINK("COLOR",C)
```

```
170 YA=96+INT(90*COS(T/56))
180 XA=128+INT(110*SIN(T/14))
190 CALL LINK("DROITE",YD,XD,YA,XA)
200 YD=YA
210 XD=XA
220 NEXT T
230 CALL KEY(5,R,E)
240 IF E=0 THEN 230
250 CALL PEEKV(-32768,X)
```

KALEIDOSCOPE

```

10 REM KALEIDOSCOPE
100 CALL LOAD(-31890,56,0)
110 CALL CLEAR
120 PRINT "          KALEIDOSCOPE":
      "          "
130 PRINT " LES COMMANDES SONT:": " C
      POUR CHANGER LA": " COULEUR
      "
140 PRINT " D,E,S,X POUR DEPLACER LE": "
      CURSEUR": " (ENTER) POUR COMMEN
      CER ET": " FINIR": "
150 INPUT "(ENTER)":R$
160 CALL LINK("BITMAP")
170 Y=20
180 X=100
190 C=32
200 CALL LINK("COLOR",C)
210 CALL LINK("POINT",Y,X)
220 X0=X
230 X=32+Y
240 Y=X0-32
250 CALL LINK("POINT",Y,X)
260 X=256-X
270 CALL LINK("POINT",Y,X)
280 Y=192-Y
290 CALL LINK("POINT",Y,X)
300 X=256-X

```

```

310 CALL LINK("POINT",Y,X)
320 X0=X
330 X=32+Y
340 Y=X0-32
350 CALL LINK("POINT",Y,X)
360 Y=192-Y
370 CALL LINK("POINT",Y,X)
380 X=256-X
390 CALL LINK("POINT",Y,X)
400 Y=192-Y
410 CALL KEY(5,R,E)
420 IF E=0 THEN 410
430 IF R<>13 THEN 460
440 CALL PEEKV(-32768,X)
450 END
460 IF R<>67 THEN 490
470 C=C+16+240*(C=240)
480 GOTO 200
490 IF R<>68 THEN 520
500 X=X+1+(X=255)
510 GOTO 210
520 IF R<>83 THEN 550
530 X=X-1-(X=8)
540 GOTO 210
550 IF R<>69 THEN 580
560 Y=Y-1-(Y=1)
570 GOTO 210
580 IF R<>88 THEN 410
590 Y=Y+1+(Y=191)
600 GOTO 210

```

LINES

```

10 REM LINES
100 CALL SCREEN(1)
110 CALL LOAD(-31890,56,0)
120 N=0
130 XD=128
140 YD=96
150 XA=211
160 YA=163
170 DXD=2
180 DYD=4
190 DXA=6
200 DYA=8
210 CALL LINK("BITMAP")
220 XD=XD+DXD
230 YD=YD+DYD
240 XA=XA+DXA
250 YA=YA+DYA
260 IF (XD>0)*(XD<256) THEN 280
270 DXD=-DXD
280 XD=XD+DXD
290 IF (XA>0)*(XA<256) THEN 310
300 DXA=-DXA
310 XA=XA+DXA

```

```

320 IF (YD>0)*(YD<192) THEN 340
330 DYD=-DYD
340 YD=YD+DYD
350 IF (YA>0)*(YA<192) THEN 370
360 DYA=-DYA
370 YA=YA+DYA
380 C=16*INT(2+14*RND)
390 CALL LINK("COLOR",C)
400 CALL LINK("DROITE",YD,XD,YA,XA)
410 N=N+1
420 IF N<80 THEN 220
430 FOR I=1 TO 20
440 CALL KEY(5,R,E)
450 IF E=0 THEN 480
460 CALL PEEKV(-32768,X)
470 END
480 NEXT I
490 N=0
500 CALL LINK("BITMAP")
510 DXD=-8+INT(16*RND)
520 DXA=-8+INT(16*RND)
530 DYD=-8+INT(16*RND)
540 DYA=-8+INT(16*RND)
550 GOTO 210

```

POLYGONES

```

10 REM POLYGONES
100 CALL LOAD(-31890,56,0)
110 CALL CLEAR
120 PRINT "          *POLYgone*":*****
130 INPUT "ANGLE ? ":DA
140 YC=96
150 XC=128
160 R=70
170 C=240
180 CALL LINK("BITMAP")
190 YD=YC
200 XD=XC+R
210 P=ATN(1)/45
220 A=A+DA

```

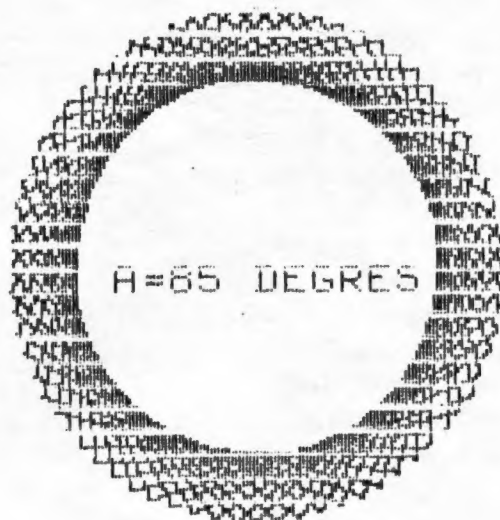
```

230 A=A-360*INT(A/360)
240 RD=A*P
250 XA=XC+R*COS(RD)
260 YA=YC+4/3*R*SIN(RD)
270 CALL LINK("DROITE",YD,XD,YA,XA)
280 IF INT(A)=0 THEN 320
290 YD=YA
300 XD=XA
310 GOTO 220
320 CALL SOUND(100,440,0)
330 CALL KEY(5,R,E)
340 IF E=0 THEN 330
350 CALL PEEKV(-32768,X)

```

Routine BITMAP appelée par CALL LINK ("BITMAP")

7D00	AORG >7D00	7D42 0420	BLWP 2>6024	7D88 03F0	
7D00 0002	DATA >0002,>01E0	7D44 6024		7D8A 0202	LI R2,>1410
7D02 01E0		7D46 0220	AI R0,>0020	7D8C 1410	
7D04 0206	DATA >0206,>03FF	7D48 0020		7D8E 0420	BLWP 2>6024
7D06 03FF		7D4A 0280	CI R0,>1900	7D90 6024	
7D08 0403	DATA >0403,>0536	7D4C 1900		7D92 0580	INC R0
7D0A 0536		7D4E 16F9	JNE >7D42	7D94 0602	DEC R2
7D0C 0603	DATA >0603,>0700	7D50 0200	LI R0,>1877	7D96 16FB	JNE >7D8E
7D0E 0700		7D52 1877		7D98 0200	LI R0,>1B00
7D10 0201	LI R1,>7D00	7D54 0201	LI R1,>2000	7D9A 1B00	
7D12 7D00		7D56 2000		7D9C 0201	LI R1,>D000
7D14 0202	LI R2,>0008	7D58 0420	BLWP 2>6024	7D9E D000	
7D16 0008		7D5A 6024		7DA0 0420	BLWP 2>6024
7D18 C031	MOV *R1+,R0	7D5C 0580	INC R0	7DA2 6024	
7D1A 0420	BLWP 2>6034	7D5E 0221	AI R1,>2000	7DA4 0202	LI R2,>F000
7D1C 6034		7D60 2000		7DA6 F000	
7D1E 0602	DEC R2	7D62 16FA	JNE >7D58	7DA8 045B	B *R11
7D20 16FB	JNE >7D18	7D64 0200	LI R0,>2000		
7D22 0200	LI R0,>1800	7D66 2000			
7D24 1800		7D68 0202	LI R2,>1800		
7D26 0202	LI R2,>0003	7D6A 1800			
7D28 0003		7D6C 0420	BLWP 2>6024		
7D2A 04C1	CLR R1	7D6E 6024			
7D2C 0420	BLWP 2>6024	7D70 0580	INC R0		
7D2E 6024		7D72 0602	DEC R2		
7D30 0580	INC R0	7D74 16FB	JNE >7D6C		
7D32 0221	AI R1,>0100	7D76 04C0	CLR R0		
7D34 0100		7D78 0202	LI R2,>03B8		
7D36 16FA	JNE >7D2C	7D7A 03B8			
7D38 0602	DEC R2	7D7C 0420	BLWP 2>6024		
7D3A 16F7	JNE >7D2A	7D7E 6024			
7D3C 0200	LI R0,>1820	7D80 0580	INC R0		
7D3E 1820		7D82 0602	DEC R2		
7D40 04C1	CLR R1	7D84 16FB	JNE >7D7C		
		7D86 0200	LI R0,>03F0		



Routine DROITE
appelée par
CALL LINK
('DROITE',YD,XD,
YA,XA)

7E2A C3CB	MOV R11,R15	7E5A 1200	DATA >1200	7E8A 070C	SET0 R12
7E2C 04C0	CLR R0	7E5C C3A0	MOV @>834A,R14	7E8C C20E	MOV R14,R8
7E2E 0201	LI R1,>0001	7E5E 834A		7E8E 620A	S R10,R8
7E30 0001		7E60 0581	INC R1	7E90 1101	JLT >7E94
7E32 0420	BLWP @>6044	7E62 0420	BLWP @>6044	7E92 1002	JMP >7E98
7E34 6044		7E64 6044		7E94 0506	NEG R6
7E36 0420	BLWP @>601C	7E66 0420	BLWP @>601C	7E96 0508	NEG R8
7E38 601C		7E68 601C		7E98 06A0	BL @>7DCE
7E3A 1200	DATA >1200	7E6A 1200	DATA >1200	7E9A 7DCE	
7E3C C2A0	MOV @>834A,R10	7E6C C360	MOV @>834A,R13	7E9C 8349	C R9,R13
7E3E 834A		7E6E 834A		7E9E 1603	JNE >7EA6
7E40 0581	INC R1	7E70 04CC	CLR R12	7EA0 838A	C R10,R14
7E42 0420	BLWP @>6044	7E72 0205	LI R5,>0001	7EA2 1601	JNE >7EA6
7E44 6044		7E74 0001		7EA4 045F	B *R15
7E46 0420	BLWP @>601C	7E76 0206	LI R6,>0001	7EA6 C30C	MOV R12,R12
7E48 601C		7E78 0001		7EA8 1103	JLT >7EB0
7E4A 1200	DATA >1200	7E7A C1CD	MOV R13,R7	7EAA A245	A R5,R9
7E4C C260	MOV @>834A,R9	7E7C 61C9	S R9,R7	7EAC 6308	S R8,R12
7E4E 834A		7E7E 1101	JLT >7E82	7EAE 10F4	JMP >7E98
7E50 0581	INC R1	7E80 1002	JMP >7E86	7EB0 A286	A R6,R10
7E52 0420	BLWP @>6044	7E82 0505	NEG R5	7EB2 A307	A R7,R12
7E54 6044		7E84 0507	NEG R7	7EB4 10F1	JMP >7E98
7E56 0420	BLWP @>601C	7E86 C1C7	MOV R7,R7	7EB6 10EE	JMP >7E94
7E58 601C		7E88 1601	JNE >7E8C		

Routine POINT
appelée par
CALL LINK
('POINT'),Y,X)

7DAA 04C0	CLR R0	7DD4 00FF		7DFE 0960	SRL R0,6
7DAC 0201	LI R1,>0001	7DD6 1B28	JH >7E28	7E00 0AB0	SLA R0,11
7DAE 0001		7DD8 0289	CI R9,>0008	7E02 A040	A R0,R1
7DB0 0420	BLWP @>6044	7DDA 0008		7E04 C009	MOV R9,R0
7DB2 6044		7DDC 1A25	JL >7E28	7E06 0240	ANDI R0,>0007
7DB4 0420	BLWP @>601C	7DDE 028A	CI R10,>00BF	7E08 0007	
7DB6 601C		7DE0 00BF		7E0A 0B03	SRC R3,0
7DB8 1200	DATA >1200	7DE2 1B22	JH >7E28	7E0C C00A	MOV R10,R0
7DBA C2A0	MOV @>834A,R10	7DE4 C009	MOV R9,R0	7E0E 0240	ANDI R0,>0007
7DBC 834A		7DE6 0930	SRL R0,3	7E10 0007	
7DBE 0581	INC R1	7DE8 C04A	MOV R10,R1	7E12 A001	A R1,R0
7DC0 0420	BLWP @>6044	7DEA 0931	SRL R1,3	7E14 0420	BLWP @>602C
7DC2 6044		7DEC 0A51	SLA R1,5	7E16 602C	
7DC4 0420	BLWP @>601C	7DEE A001	A R1,R0	7E18 E043	SOC R3,R1
7DC6 601C		7DF0 0220	AI R0,>1800	7E1A 0420	BLWP @>6024
7DC8 1200	DATA >1200	7DF2 1800		7E1C 6024	
7DCA C260	MOV @>834A,R9	7DF4 04C1	CLR R1	7E1E 0220	AI R0,>2000
7DCC 834A		7DF6 0420	BLWP @>602C	7E20 2000	
7DCE 0203	LI R3,>8000	7DF8 602C		7E22 C042	MOV R2,R1
7DD0 8000		7DFA 0951	SRL R1,5	7E24 0420	BLWP @>6024
7DD2 0289	CI R9,>00FF	7DFC C00A	MOV R10,R0	7E26 6024	
				7E28 045B	B *R11

Routine COLOR
appelée par
CALL LINK
(*COLOR*,C)

Table REF/DEF
Le pointeur de
début de table est
en >701E

```
7EB8 04C0 CLR R0
7EBA 0201 LI R1,>0001
7EBC 0001
7EBE 0420 BLWP @>6044
7EC0 6044
7EC2 0420 BLWP @>601C
7EC4 601C
7EC6 1200 JLE >7EC8
7EC8 C0A0 MOV @>834A,R2
7ECA 834A
7ECC 06C2 SWPB R2
7ECE 045B B *R11
```

```
7FD0 AORG >7FD0
7FD0 434F TEXT 'COLOR '
7FD2 4C4F
7FD4 5220
7FD6 7EB8 DATA >7EB8
7FD8 4452 TEXT 'DROITE'
7FDA 4F49
7FDC 5445
7FDE 7E2A DATA >7E2A
7FE0 504F TEXT 'POINT '
7FE2 494E
```

```
7FE4 5420
7FE6 7DAA DATA >7DAA
7FE8 4249 TEXT 'BITMAP'
7FEA 544D
7FEC 4150
7FEE 7D10 DATA >7D10
701E AORG >701E
701E 7FD0 DATA >7FD0
END
```

Le train sifflera trois fois...

Denise Amrouche, Gilbert Arribet et Jean Sahal

...dès qu'il sera en vue de la bourgade dont vous êtes le chef de gare. Mais les choses se compliquent lorsque deux trains arrivent en même temps et doivent se croiser. En effet, à cet endroit, la gare est traversée par une voie unique et vous disposez d'une voie de garage tout juste assez grande pour contenir la locomotive et un autre wagon, et rien de plus. Pour corser le tout, les chauffeurs de locomotive ne comprennent pas le français... (vous avez compris, c'est parce qu'ils veulent que vous perfectionniez votre anglais !).

Votre mission, si vous l'acceptez, consistera donc à diriger la manœuvre compliquée qui permettra à chaque train de poursuivre sa route. Pour cela, vous envoyez des messages successifs

aux chefs de train concernés. Dans la mesure où votre message est rationnel et compréhensible, il est exécuté et vous assistez à l'opération. Dans le cas contraire, rien ne se passe et vous recevez un message d'explication qui apparaît sur votre téléscrip-teur. Toute erreur est comptabilisée. Si vous provoquez une collision, vous êtes renvoyé. Si vous faites très peu d'erreurs, courez vite vous faire embaucher par les chemins de fer britanniques.

Une recommandation utile : lorsque vous envoyez un message, assurez-vous que vous ne vous êtes pas trompé de destinataire. C'est LEW (avec un "L" comme "Londres") qui conduit la locomotive qui se dirige vers Londres; BOB (avec un "B" comme "Bristol") qui a la charge de la

locomotive à destination de Bristol. Pour ceux qui ont des lacunes en géographie, Londres est à l'est, Bristol est à l'ouest.

Du point de vue pédagogique, ce programme permet de comprendre in situ, un point de grammaire anglaise délicat : l'emploi de certains verbes de mouvement, ceci au moyen d'une simulation rendue possible grâce aux nombreuses possibilités graphiques et sonores du TI-99 associé au Basic Etendu. Rien n'y manque : le teuf teuf de la locomotive, le crépitement des téléscrip-teurs, le fracas éventuel en cas de collision et même le leit-motiv du célèbre film avec Gary Cooper et Grace Kelly.

□

```

100 !      DEVENEZ CHEF DE GARE A LONDRES
110 !
120 !
130 !      PAR DENISE AMROUCHE ET JEAN SAH
AL
140 !
150 !
160 !      TI 99/4A BASIC ETENDU
170 !
180 CALL CHAR(136,"0F18103000783033337FF
FFF7F3B7BF900B040003F2424243CFFFFFFFDE
E8C")
190 CALL CHAR(140,"00000000007F7F7F7F7F
F7FFF9E1E0C000000000FEFEFEFEFEFEFEFF797
830")
200 CALL CHAR(120,"0609003CE424243FFFFF
FFFFFFB79303C409E0C0CCCFEFFFFFFEFCDEDFC08
000")
210 CALL CHAR(124,"0815123000783033337FF
FFF7F3B79F84080073C2424273FFFFFFFDECC8
")
220 CALL CHAR(116,"000000000077F7F7F7F7
F7F7F5E8C80000000003EFEFEFEFEFEFEFFDF8783")
230 CALL CHAR(108,"917139218120CC2C40E2F
A20002DCDC0C3C3C690101010D00101F903973E8
C88")
240 CALL CLEAR :: CALL SCREEN(3):: CALL
MAGNIFY(3)
250 DISPLAY AT(15,1):" THE STATION MAST
ER GAME"
260 GOSUB 470 :: CALL SCREEN(8)
270 DISPLAY AT(17,1):"YOU ARE THE STATIO
N MASTER":"OF A SMALL AND QUIET RAILWAYS
TATION. BY THE WAY WHAT IS"
280 DISPLAY AT(20,1):"YOUR NAME?" :: ACC
EPT AT(20,12)SIZE(9)VALIDATE(UALPHA)BEEP
:NOM$ :: DISPLAY AT(20,1):" "
290 MM$=NOM$&" - MESSAGE RECIEVED - PRES
S ANY KEY TO START THE MOTION."
300 DISPLAY AT(15,1):"":"SOMETIMES YOU S
EE THE TRAIN COMING FROM BRISTOL ON ITS
WAY TO LONDON":""
310 GOSUB 1160 :: CALL DEL(400):: GOSUB
1400 :: CALL LHAU :: GOSUB 1200 :: CALL
DELSprite(ALL)
320 DISPLAY AT(15,1):"":"SOMETIMES YO
U SEE THE LONDONTRAIN GOING TO BRISTOL":
""
330 GOSUB 1160 :: CALL DEL(600):: GOSUB
1430 :: CALL BBAS :: GOSUB 1200 :: CALL
DELSprite(ALL):: CALL TOUCH
340 DISPLAY AT(15,1):"":"NOTHING REALLY
HAPPENS ... YET...OCCASIONALLY YOU MAY
HAVE SOME PROBLEMS":"":"HERE IS ONE EXA
MPLE...":""
350 CALL DEL(200):: GOSUB 1400 :: GOSUB
1430 :: CALL LHAU :: CALL BBAS :: GOSUB
1160
360 GOSUB 1450
370 CALL DEL(150):: CALL BLAN :: DISPLAY
AT(17,1):"GOOD JOB ";NOM$:"THANKS HEAVE
NS!":"WE HAD A NARROW ESCAPE" :: CALL DE
L(200):: CALL TOUCH
380 GOSUB 1570
390 DISPLAY AT(15,1):"CONGRATULATIONS NO
W YOU ARE READY TO SEND MESSAGES OF YO
UR OWN WITHOUT ANY HELP. WELL GO AHEAD.
"
400 GOSUB 1400 :: GOSUB 1430 :: ER=0 ::
CALL TOUCH
410 FOR MOVE=1 TO 10
420 DISPLAY AT(15,1)SIZE(11):"MOVE NR."&
STR$(MOVE)&": " :: GOSUB 1680 :: ON MOVE
GOSUB 700,750,790,840,880,930,980,1020,1
070,1120
430 NEXT MOVE
440 CALL BLAN :: DISPLAY AT(15,1):NOM$:"
YOU 'VE DONE IT!..."
450 GOSUB 1290 :: CALL CLEAR :: CALL DEL
SPRITE(ALL):: STOP
460 ! CONFECTION DE LA VOIE
470 CALL COLOR(13,2,4,11,2,4,10,2,4)
480 CALL CHAR(128,"00010FFCE0031FF83EF08
10F7CE00000073EF08000000000000000000000
007")
490 CALL CHAR(112,"00FF0F7CFF031FF83EF08
00FFCE00000000FF0000FF0000070")
500 CALL CHAR(132,"00000000031FF8C0C0000
00000000000000FF0000FF00000000FF0000FF1EF
8C")
510 CALL CHAR(104,"00000000011F3D6CCCF
FFFF")
520 CALL HCHAR(1,1,115,416)
530 FOR I=0 TO 9
540 CALL HCHAR(13-I,1+3*I,128):: CALL HC
HAR(13-I,2+3*I,129):: CALL HCHAR(13-I,3+
3*I,130):: CALL HCHAR(13-I,4+3*I,133)
550 CALL HCHAR(12-I,2+3*I,131):: CALL HC
HAR(12-I,3+3*I,132):: NEXT I
560 CALL HCHAR(3,31,128):: CALL HCHAR(3,
32,129):: CALL HCHAR(2,32,131)
570 CALL HCHAR(7,11,104):: CALL HCHAR(8,
11,105)
580 CALL HCHAR(8,12,134,2):: CALL HCHAR(
8,14,114):: CALL HCHAR(8,15,135):: CALL
HCHAR(8,16,112):: CALL HCHAR(8,17,113)
590 RETURN
600 !TRADUCTION DES MESSAGES EN MOUVEMEN
T
610 !0

```



```

620 CALL ORDR(REP$,"LEW","PUSH","TRAIN",
"BACK",TEST,RF$,ER):: IF TEST=0 THEN 640
630 CALL BANDE(24,MM$,1):: CALL LBAS ::
CALL LOCALS(1,75):: CALL ALT :: RETURN
640 GOSUB 1730 :: GOTO 620
650 !00
660 CALL ORDR(REP$,"BOB","PUSH","TRAIN",
"BACK",TEST,RF$,ER):: IF TEST=0 THEN 680
670 CALL BANDE(24,MM$,1):: CALL BHAU ::
CALL LOCALI(4,18):: CALL ALT :: CALL BLA
N :: RETURN
680 GOSUB 1730 :: GOTO 660
690 !1
700 CALL ORDR(REP$,"BOB","PULL","THE YEL
LOW VAN","INTO THE SIDING",TEST,RF$,ER):
: IF TEST=0 THEN 730
710 CALL BANDE(24,MM$,1):: FOR W=4 TO 5
:: CALL BAS(W):: NEXT W :: CALL LOCALS(4
,45):: CALL PATTERN(#4,136):: CALL GAU(4
)
720 CALL LOCALS(5,45):: CALL PATTERN(#5,
140):: CALL GAU(5):: CALL XLOCI(4,96)::
CALL ALT :: CALL BLAN :: RETURN
730 GOSUB 1730 :: GOTO 700
740 !2
750 CALL ORDR(REP$,"LEW","PULL","TRAIN",
"PAST THE SIDING",TEST,RF$,ER):: IF TEST
=0 THEN 770
760 CALL BANDE(24,MM$,1):: CALL LHAU ::
CALL LOCALI(1,35):: CALL ALT :: CALL BLA
N :: RETURN
770 GOSUB 1730 :: GOTO 750
780 !3
790 CALL ORDR(REP$,"LEW","PULL","THE YEL
LOW VAN ","OUT OF THE SIDING",TEST,RF$,E
R):: IF TEST=0 THEN 820
800 CALL BANDE(24,MM$,1):: CALL DTE(5)::
CALL LHAU :: CALL XLOCS(1,176):: CALL P
ATTERN(#5,116):: CALL HAU(5)
810 CALL LOCALI(1,28):: CALL ALT :: CALL
BLAN :: RETURN
820 GOSUB 1730 :: GOTO 790
830 !4
840 CALL ORDR(REP$,"LEW","PUSH"," ","BAC
K",TEST,RF$,ER):: IF TEST=0 THEN 860
850 CALL BANDE(24,MM$,1):: CALL BAS(5)::
CALL LBAS :: CALL LOCALS(1,60):: CALL A
LT :: CALL BLAN :: RETURN
860 GOSUB 1730 :: GOTO 840
870 !5
880 CALL ORDR(REP$,"BOB","MOVE"," ","OUT
OF THE SIDING",TEST,RF$,ER):: IF TEST=0
THEN 910
890 CALL BANDE(24,MM$,1):: CALL DTE(4)::
CALL XLOCS(4,126):: CALL PATTERN(#4,124
):: CALL HAU(4):: CALL LOCALI(4,12):: CA
LL ALT
900 CALL BLAN :: RETURN
910 GOSUB 1730 :: GOTO 880
920 !6
930 CALL ORDR(REP$,"BOB","PULL","THE BLU
E VAN ","INTO THE SIDING",TEST,RF$,ER)::
IF TEST=0 THEN 960
940 CALL BANDE(24,MM$,1):: CALL BAS(6)::
CALL BAS(4):: CALL LOCALS(4,45):: CALL
PATTERN(#4,136):: CALL GAU(4):: CALL LOC
ALS(6,45)
950 CALL PATTERN(#6,140):: CALL GAU(6)::
CALL XLOCI(6,114):: CALL ALT :: CALL BL
AN :: RETURN
960 GOSUB 1730 :: GOTO 930
970 !7
980 CALL ORDR(REP$,"LEW","PULL"," ","PAS
T THE SIDING",TEST,RF$,ER):: IF TEST=0 T
HEN 1000
990 CALL BANDE(24,MM$,1):: CALL LHAU ::
CALL HAU(5):: CALL LOCALI(1,27):: CALL A
LT :: CALL BLAN :: RETURN
1000 GOSUB 1730 :: GOTO 980
1010 !8
1020 CALL ORDR(REP$,"LEW","PULL","THE BL
UE VAN ","OUT OF THE SIDING",TEST,RF$,ER
):: IF TEST=0 THEN 1050
1030 CALL BANDE(24,MM$,1):: CALL DTE(6):
: CALL LHAU :: CALL HAU(5):: CALL XLOCS(
1,199):: CALL PATTERN(#6,116):: CALL HAU
(6)
1040 CALL LOCALI(1,10):: CALL ALT :: CAL
L BLAN :: RETURN
1050 GOSUB 1730 :: GOTO 1020
1060 !9
1070 CALL ORDR(REP$,"BOB","MOVE"," ","OU
T OF THE SIDING",TEST,RF$,ER):: IF TEST=
0 THEN 1100
1080 CALL BANDE(24,MM$,1):: CALL DTE(4):
: CALL XLOCS(4,128):: CALL PATTERN(#4,12
4):: CALL HAU(4):: CALL LOCALI(4,40):: C
ALL ALT
1090 CALL BLAN :: RETURN
1100 GOSUB 1730 :: GOTO 1070
1110 !10
1120 CALL ORDR(REP$,"BOB","PULL"," ","AW
AY",TEST,RF$,ER):: IF TEST=0 THEN 1140
1130 CALL BANDE(24,MM$,1):: CALL BAS(4):
: CALL BAS(6):: CALL BAS(5):: CALL LOCAL
S(4,70):: CALL ALT :: CALL BLAN :: RETUR
N
1140 GOSUB 1730 :: GOTO 1120
1150 ! SIFFLET

```

```

1160 CALL SOUND(100,770,5,800,5,750,5)::
CALL DEL(50)
1170 CALL SOUND(1500,770,0,800,0,750,0)
1180 RETURN
1190 ! TEUF-TEUF
1200 D=190 :: V=0
1210 FOR I=1 TO 6
1220 IF I=1-3*(I=4) THEN D=D*2
1230 CALL SOUND(D,-5,V,110,V)
1240 IF I=1-3*(I=4) THEN D=D/2
1250 NEXT I
1260 IF D<20 THEN RETURN
1270 D=D/1.25 :: V=V+.5 :: GOTO 1210
1280 ! LE TRAIN SIFFLERA TROIS FOIS
1290 DATA 165,1,220,1,247,1,277,1,220,1,
294,1,277,1,247,1,220,6
1300 DATA 220,1,247,1,277,1,247,1,220,3,
185,4,247,5,185,1,247,1,277,1
1310 DATA 294,1,247,1,330,1,294,1,277,4,
370,4,349,8,330,9
1320 DATA 165,1,220,1,247,1,277,1,220,1,
294,1,277,1,247,1,220,6
1330 DATA 220,1,220,1,247,1,277,1,220,1,
330,1,220,1,370,7,370,1,370,1,415,1,440,
2
1340 DATA 415,1,370,1,330,1,220,6,220,1,
220,1,247,1,277,1,330,4,220,1
1350 DATA 220,1,277,1,330,2,330,1,277,1,
220,2,247,2,220,6
1360 FOR I=1 TO 66 :: READ F1,D
1370 CALL SOUND(D*275,F1,0,F1*1.94,15,F1
*.994,3)
1380 NEXT I :: RETURN
1390 ! CREATION DU TRAIN ALLANT A LONDRE
S
1400 CALL SPRITE(#1,120,2,78,31,#2,116,7
,84,16,#3,116,7,89,1)
1410 RETURN
1420 ! TRAINS VENANT DE LONDRES
1430 CALL SPRITE(#4,124,02,17,213,#5,116
,11,13,229,#6,116,5,8,245):: RETURN
1440 ! TESTER DANGER D'EXPLOSION
1450 CALL BLAN :: CALL SOUND(80,770,5,80
0,5,750,5):: DISPLAY AT(17,1):"QUICK ";M
EC$:"PRESS ANY TOUCH TO AVOID ":" A
D I S A S T E R "
1460 CALL COINC(#1,#4,15,CC):: CALL KEY(
0,K,S):: IF S<>0 THEN 1490
1470 IF CC=-1 THEN 1480 :: GOTO 1450
1480 CALL EXPLO :: GOTO 1510
1490 CALL ALT :: RETURN
1500 CALL KEY(0,K,S):: IF S=0 THEN 1500
ELSE 1470 :: RETURN
1510 DISPLAY AT(15,1):"W A N T E D"
: "D E A D O R A L I V E": :NOM$: "TH

```

```

E STATION MASTER": "WHO CAUSED A DISASTE
R"
1520 CALL TOUCH :: CALL CLEAR :: CALL DE
LSprite(ALL)
1530 PRINT "V A C A N C Y " : : : : "STATI
ON MASTER REQUIRED HIGH": "SALARY. NOTH
ING SPECIAL TO ": "DO, JUST WATCH THE T
RAINS": :
1540 PRINT "PASS BY AND OCCASIONALLY USE
": "A SIDING PROPERLY. PLENTY OF": "TI
ME FREE TO PLAY WITH YOUR": :
1550 PRINT "PERSONNAL COMPUTER.APPLY TO"
: "BRITISH RAILWAYS - LONDON": : : : :
: CALL TOUCH
1560 CALL CLEAR :: CALL SCREEN(3):: REST
ORE :: GOTO 260
1570 DISPLAY AT(15,1):"NOW YOUR WORK BEG
INS.SEND TO EACH ENGINE DRIVER MESSAGES A
BOUT HOW TO USE THE SIDING IN ORDER TO P
ASS. IT IS BIG"
1580 DISPLAY AT(19,1):"ENOUGH TO HOLD ON
E ENGINE AND ONE WAGON ONLY." :: CALL
TOUCH
1590 DISPLAY AT(15,6):"NOW LET'S SEE IF
YOU CAN": "SEND A MESSAGE CORRECTLY," : "TH
ERE WILL BE SOME WORDS ON THE SCREEN TO
HELP YOU"
1600 CALL TOUCH
1610 DISPLAY AT(1,1)SIZE(27):"INTO/OUT O
F/PAST THE SIDING"
1620 DISPLAY AT(2,1)SIZE(24):"MOVE-PUSH-
PULL-AWAY-BACK"
1630 DISPLAY AT(3,1)SIZE(14):"THE/YOUR T
RAIN"
1640 DISPLAY AT(4,1)SIZE(14):"THE YELLOW
VAN"
1650 DISPLAY AT(5,1)SIZE(12):"THE BLUE V
AN"
1660 DISPLAY AT(15,1):"TELL LEW TO MOVE
BACK A BIT" :: GOSUB 1690 :: GOSUB 610
1670 DISPLAY AT(15,1):"NOW TELL BOB THE
SAME." :: GOSUB 1690 :: GOSUB 650 :: GOT
O 390
1680 DISPLAY AT(15,12):"TYPE YOUR MESSAG
E"
1690 DISPLAY AT(16,1):"USE MOSTLY THE WO
RDS ON THE SCREEN; 2 LINES AVAILABLE P
RESS ENTER AFTER EACH LINE NO SPLIT EXPR
SSIONS:"
1700 DISPLAY AT(21,1):"TO: ":"(CONT.): "
: "
1710 ACCEPT AT(21,04)VALIDATE(UALPHA)SIZ
E(-28)BEEP:REP1$
1720 ACCEPT AT(22,09)VALIDATE(UALPHA)SIZ
E(-28)BEEP:REP2$ :: REP$=REP1$&" "&REP2$

```

```

:: RETURN
1730 CALL BANDE(24,NOM$&RF$,1):: GOSUB 1
710 :: RETURN
1740 !SERIE DE CALL CECI OU CELA
1750 SUB EXPLO :: CALL ALT :: CALL BLAN
:: CALL SOUND(1000,-6,0,200,0):: FOR SC=
1 TO 15 :: FOR T=1 TO 5 :: NEXT T :: CAL
L SCREEN(SC):: NEXT SC
1760 CALL SCREEN(10):: CALL PATTERN(#1,1
08,#2,108,#4,108,#5,108)
1770 XLOC=INT(RND*15+120):: YLOC=INT(RND
*150+1):: CALL LOCATE(#1,YLOC,XLOC)
1780 XLOC=INT(RND*15+150):: YLOC=INT(RND
*150+1):: CALL LOCATE(#4,YLOC,XLOC)
1790 XLOC=INT(RND*15+120):: YLOC=INT(RND
*150+1):: CALL LOCATE(#2,YLOC,XLOC)
1800 XLOC=INT(RND*20+100):: YLOC=INT(RND
*150+10):: CALL LOCATE(#5,YLOC,XLOC)
1810 SUBEND
1820 SUB TOUCH
1830 DISPLAY AT(24,1):"PRESS ANY KEY TO
GO ON"
1840 CALL KEY(0,K,S):: IF S<>1 THEN 1840
1850 CALL BLAN
1860 SUBEND
1870 !MOUVEMENT DES TRAINS
1880 SUB BAS(W)!VERS BRISTOL
1890 CALL MOTION(#W,2,-6):: SUBEND
1900 SUB LBAS !TR VERS LDRES DESCEND
1910 FOR W=3 TO 1 STEP -1 :: CALL MOTION
(#W,2,-6):: NEXT W :: SUBEND
1920 SUB BBAS !TRAIN VERS BRISTOL DESCEN
D
1930 FOR W=4 TO 6 :: CALL MOTION(#W,2,-6
):: NEXT W :: SUBEND
1940 !
1950 SUB HAU(W)!VERS LONDRES
1960 CALL MOTION(#W,-2,6):: SUBEND
1970 SUB LHAU !TR DE LONDRES MONTE
1980 FOR W=1 TO 3 :: CALL MOTION(#W,-2,6
):: NEXT W :: SUBEND
1990 SUB BHAU !TR DE BRISTOL MONTE
2000 FOR W=4 TO 6 :: CALL MOTION(#W,-2,6
):: NEXT W :: SUBEND
2010 !
2020 SUB DTE(W)!SORTIE DE SD
2030 CALL MOTION(#W,0,6):: SUBEND
2040 SUB GAU(W)!ENTRE DS SD
2050 CALL MOTION(#W,0,-6):: SUBEND
2060 SUB STO(W)!W=NUMERO DU SPRITE
2070 CALL MOTION(#W,0,0):: SUBEND
2080 SUB ALT !ARRET TOTAL
2090 FOR W=1 TO 6 :: CALL MOTION(#W,0,0)
:: NEXT W :: SUBEND

```

```

2100 SUB BLAN :: DISPLAY AT(14,1):"" :
: : : : : : : : SUBEND
2110 SUB DEL(D):: FOR T=1 TO D :: NEXT T
:: SUBEND
2120 SUB BANDE(L1,M$,VIT)
2130 BA$=RPT$(" ",28)&M$
2140 FOR I=1 TO LEN(BA$):: DISPLAY AT(L1
,1):SEG$(BA$,I,28):: CALL SOUND(VIT,4000
0,0):: CALL KEY(0,K,S):: IF S<>0 THEN GO
SUB 1160 :: GOTO 2160
2150 NEXT I :: GOTO 2140
2160 DISPLAY AT(24,1):"" :: SUBEND
2170 SUB LOCALS(W,XX)
2180 CALL POSITION(#W,X,Y):: IF X=XX TH
EN 2190 ELSE 2180
2190 SUBEND
2200 SUB LOCALI(W,XX)
2210 CALL POSITION(#W,X,Y):: IF X=XX TH
EN 2220 ELSE 2210
2220 SUBEND
2230 SUB XLOCI(W,XX)
2240 CALL POSITION(#W,X,Y):: IF Y=XX TH
EN 2250 ELSE 2240
2250 SUBEND
2260 SUB XLOCS(W,XX)
2270 CALL POSITION(#W,X,Y):: IF Y=XX TH
EN 2280 ELSE 2270
2280 SUBEND
2290 SUB ORDR(REP$,DR$,VE$,CPL$,ADV$,TES
T,RF$,ER)
2300 DR=POS(REP$,DR$,1)
2310 VE=POS(REP$,VE$,1)
2320 CPL=POS(REP$,CPL$,1)
2330 ADV=POS(REP$,ADV$,1)
2340 TEST=DR*VE*CPL*ADV :: IF TEST=0 THE
N ER=ER+1
2350 ERR$="BY THE WAY, YOU MADE "&STR$(ER
)&" MISTAKE"&CHR$(-83*(ER>1))&". "
2360 RF$=" USE THE ABOVE EXPRESSIONS UNS
PLIT MAINLY: "&ADV$&" - "&VE$&" - "& DR
IVER'S NAME . "&ERR$&"- PRESS ANY KEY TO
GET ANOTHER CHANCE - "
2370 SUBEND

```


Comptes familiaux

Jean-Philippe Guillemant

Présentation

Ce programme de comptes familiaux offre la possibilité de gérer avec le TI, sur disquette, mois par mois et rubrique par rubrique, les recettes et dépenses de la famille, et cela pendant 17 années non obligatoirement consécutives.

A partir des entrées effectuées, le TI pourra afficher l'état des comptes, les bilans mensuels ou annuels, l'évolution des recettes et dépenses sur une année ou l'évolution annuelle d'une rubrique choisie parmi les 18 disponibles en recettes ou les 18 disponibles en dépenses.

Configuration

Il est nécessaire de disposer, avec le TI-99/4A, du module "Basic Etendu", de l'extension de mémoire et d'un lecteur de disquettes. Une imprimante serait bienvenue, mais la routine correspondante n'est pas développée (voir paragraphe "Impression des résultats").

Utilisation

Appel

Une disquette complète est réservée à la gestion des comptes, ce qui permet de nommer le programme "DSK1.LOAD" de manière à ce qu'il soit chargé automatiquement à la mise en route du Basic étendu.

Menu principal

Après une phase d'initialisation, pendant laquelle les variables sont préparées et la structure des comptes lue en mémoire, le menu principal est affiché. 7 options sont proposées; il en existe une huitième qui n'est pas affichée car elle est dangereuse : il s'agit de l'option 0 (voir paragraphe

suisant). On choisit une option en tapant le numéro correspondant suivi par <ENTER>.

A la fin de chaque fonction, le programme revient au menu principal.

Quand on revient dans une fonction déjà exécutée, y compris dans le menu principal, les réponses tapées précédemment sont affichées à nouveau. Il suffit de taper <ENTER> pour les prendre en compte à nouveau.

Initialisation (option 0)

ATTENTION : cette option est dangereuse; c'est pourquoi elle n'apparaît pas dans le menu. Elle est essentiellement utile en période de test. Elle permet de réinitialiser les fichiers utilisés par le programme. Toutes les informations précédemment saisies sont perdues dès que l'on a répondu "0" à la demande de confirmation.

Entrée des rubriques (option 1)

On dispose de 18 rubriques en recettes et 18 en dépenses. Chaque rubrique a un nom de 18 caractères et une abréviation de 4 caractères (utilisée dans les bilans, voir paragraphe "Bilan comptable").

Le programme affiche les 10 premiers caractères de chaque rubrique. Les noms des rubriques non définies sont remplacées par des points. Chaque rubrique est numérotée.

Il demande ensuite le type de rubrique à créer ou à modifier ("1" pour recettes ou "2" pour dépenses), puis le numéro de la rubrique concernée (1 à 18). Si la rubrique est déjà définie, il affiche son nom et son abréviation. On peut alors entrer un nouveau nom (sur 18 caractères maximum) et une nouvelle abréviation (sur 4 caractères maximum). Si on ne rentre aucun caractère pour le nom, la rubrique est annulée et l'abréviation n'est pas demandée.

On peut modifier plusieurs rubriques successivement en répondant "0" à la question "ENCORE (O/N)". Quand on répond "N", le fichier des rubriques est mis à jour et le programme revient au menu principal.

Les options suivantes (options 2 à 6) ne peuvent pas être appelées tant que le fichier des rubriques n'est pas créé.

Entrée des comptes (option 2)

Le programme commence par demander l'année (sur 2 chiffres), le mois (de 1 à 12) et le type de rubriques ("1" pour Recettes ou "2" pour Dépenses) concernés. Après avoir chargé en mémoire les comptes correspondants (s'ils existent), il demande si on désire modifier le montant d'une rubrique ou de toutes.

Si on répond "0" à cette question, il demande la rubrique concernée après avoir affiché, pour mémoire, le numéro et le nom (les 10 premiers caractères) des rubriques. On peut alors mettre à jour cette rubrique. Le nouveau montant est affiché immédiatement. Quand on ne veut pas modifier le montant, il suffit de taper <ENTER>.

Ces modifications peuvent être effectuées tant que l'on répond "0" à la question "ENCORE (O/N)". On enregistre les modifications et on revient au menu principal en répondant "N" à cette question.

Si on a répondu "N" à la question "MODIFICATION D'UNE RUBRIQUE", le programme affiche toutes les rubriques pour ce mois et ce type et met le curseur en face du premier montant. On peut le laisser tel quel en tapant <ENTER> ou le modifier. Quand toutes les rubriques ont été passées en revue, si on répond "0" à la question "CORRECT (O/N)", les nouveaux comptes sont enregistrés et on revient au menu principal, sinon on recommence la vérification.

Etat des comptes (option 3)

Cette fonction indique, sous forme de tableau par année et par mois, les comptes déjà saisis. La lettre "R" est affichée si les recettes existent, la lettre "D" si les dépenses existent, le symbole "RD" si recettes et dépenses existent ou un tiret si les comptes correspondants n'ont pas été créés.

On revient au menu principal en appuyant sur la touche <ENTER> (voir paragraphe "Impression des résultats").

Bilan comptable (option 4)

La fonction demande si on souhaite obtenir le bilan correspondant à un mois donné ou à une année complète. Il faut répondre "1" pour un bilan mensuel (réponse par défaut) ou "2" pour un bilan annuel. Elle demande ensuite l'année désirée et, éventuellement, le mois.

Après avoir fait les calculs correspondants, elle affiche l'ensemble des recettes et dépenses pour le mois ou l'année concerné, ainsi que les totaux et le solde. C'est dans cette fonction que sont utilisées les abréviations des noms de rubriques. On revient au menu principal en appuyant sur <ENTER> (voir paragraphe "Impression des résultats").

Evolution annuelle (option 5)

Cette fonction permet de visualiser l'évolution, mois par mois, des recettes et des dépenses d'une année donnée (même si les comptes de l'année ne sont pas complets).

Après avoir demandé l'année concernée, le système affiche un tableau indiquant le total des recettes et le total des dépenses de chaque mois, ainsi que les totaux et le solde annuels.

Après que l'on ait appuyé sur <ENTER> (voir paragraphe "Impression des résultats"), le système affiche les mêmes résultats, mais sous forme d'un graphique en bâtons verticaux, les recettes vers le haut de l'écran et les dépenses vers le bas. Le graphique est gradué en dixièmes du montant le plus

élevé et la valeur de l'unité de graduation est indiquée sous le graphique. On revient au menu principal en appuyant sur <ENTER> (voir paragraphe "Impression des résultats").

Evolution d'une rubrique (option 6)

Il est possible, grâce à cette fonction, de connaître l'évolution, pour une année donnée, d'une rubrique quelconque. Le programme demande tout d'abord le type de rubrique ("1" pour recettes ou "2" pour dépenses) et l'année désirée. Après avoir affiché, pour mémoire, les numéros et libellés des rubriques (sur 10 caractères), il demande le numéro de la rubrique que l'on souhaite voir évoluer.

Le système affiche alors, mois par mois, les montants associés à la rubrique choisie, ainsi que la moyenne mensuelle et le total annuel.

Quand on appuie sur <ENTER> (voir paragraphe "Impression des résultats"), le système montre l'évolution de la rubrique sous la forme d'un graphique en bâtons horizontaux. Ce graphique est gradué en vingtièmes du montant le plus élevé et la valeur de l'unité de graduation est indiquée sous le graphique. On revient au menu principal en tapant <ENTER> (voir paragraphe "Impression des résultats").

Sortie du programme (option 7)

Cette fonction permet bien évidemment de terminer la gestion des comptes. Il est obligatoire de sortir par cette option car l'action des touches <BREAK> et <QUIT> est interdite.

Impression des résultats

Ce programme est particulièrement intéressant si on dispose d'une imprimante pour conserver les tableaux. A défaut d'en avoir, seul l'appel de l'impression est actuellement prévu.

Cela doit se faire par appui sur les touches "P" et <ENTER> après l'affichage des tableaux (options 3, 4, 5 et 6). La routine correspondante s'appelle "IMP_ECR" et n'a qu'à effectuer une recopie d'écran.

Erreurs

En cas d'erreur pendant l'exécution, le programme affiche le code d'erreur (cf. l'appendice N du manuel Basic Etendu) et le numéro de ligne et attend que l'on ait appuyé sur <ENTER> pour se terminer.

Le programme

Structure du programme

Le programme est organisé en modules distincts pour chaque fonction, respectivement :

INITIALISATION	lignes	100 - 530
MENU PRINCIPAL	lignes	540 - 720
ERREURS	lignes	730 - 760
OPTION 0	lignes	770 - 850
OPTION 1	lignes	860 - 1020
OPTION 2	lignes	1030 - 1550
OPTION 3	lignes	1560 - 1680
OPTION 4	lignes	1690 - 2000
OPTION 5	lignes	2010 - 2360
OPTION 6	lignes	2370 - 2750
IMPRESSION	lignes	2760 - 2810

chacun de ces modules étant indépendant des autres pour des facilités de programmation et de corrections.

Description des fichiers

Fichier des rubriques

Le nom de ce fichier est rangé dans la variable F_RUB\$. Il est défini en enregistrements de longueur variables de 48 octets chacun. Il contient 18 enregistrements. La structure de ces enregistrements est la suivante :

- libellé recette 19 octets
- abréviation recette 5 octets
- libellé dépense 19 octets
- abréviation dépense 5 octets

pour chacune des 18 rubriques possibles.

Fichiers des comptes

Le nom de ce fichier est rangé dans la variable F_MONT\$. Il est défini en enregistrements de longueur fixe de manière à pouvoir y accéder en accès direct. La longueur des

enregistrements est de 162 octets, soit 18 nombres en représentation interne (9 octets chacun). Le contenu des enregistrements est différents selon leur position :

• **Enregistrement 0**

Il permet de repérer les années existantes. Les 17 premiers nombres contiennent soit 0, soit un numéro d'année créée. Si l'année existe, sa position dans cet enregistrement permet de pointer sur les enregistrements concernés (voir "Enregistrements 1 à 34"). Ceci permet de créer les années dans le désordre.

Le dernier nombre donne le nombre d'enregistrements existant dans le fichier, de manière à pouvoir en ajouter.

• **Enregistrements 1 à 34**

Il existe deux enregistrements de ce type par année. Le premier pointe sur les recettes de l'année, le deuxième sur les dépenses. Seuls les 12 premiers nombres sont utilisés. Ils contiennent respectivement pour chaque mois de l'année soit 0, soit le numéro de l'enregistrement des comptes correspondants si ceux-ci existent. Ceci permet de créer les comptes mensuels dans le désordre.

• **Enregistrements 35 et suivants**

Ils contiennent les montants saisis, soit en recettes, soit en dépenses, dans l'ordre des numéros de rubriques.

Redéfinition des caractères

Code 91 : dessin diagramme option 5, recettes.

Code 92 : double souligné.

Code 93 : dessin diagramme option 5, dépenses.

Code 94 : dessin diagramme option 6.

Code 95 : souligné et cadre tableaux.

Code 124 : cadre tableaux.

Code 126 : symboles recettes/

dépenses, option 3.

Codes 128 à 135 : bâtons recettes option 5, bâtons option 6.

Codes 136 à 143 : bâtons dépenses option 5.

Variables

Toutes les variables sont réservées au début du programme.

Variables générales

Elles sont utilisées dans l'ensemble du programme.

• **FRUB** Vrai (-1) si le fichier rubriques existe. Faux (0) sinon (valeur initiale).

• **RECN** Nombre d'enregistrements du fichier comptes (vaut 35 à l'initialisation).

• **OFF_AN** Déplacement de l'année dans l'enregistrement 0 du fichier comptes; permet d'accéder aux enregistrements recettes et dépenses.

• **F_RUB\$** Contient le nom du fichier des rubriques.

• **F_MONT\$** Contient le nom du fichier des comptes.

A\$, I, J et K sont des variables temporaires.

Variables locales

Pour plus de clarté, les variables suivantes sont affectées à chaque module en ajoutant le numéro du module au nom de la variable; cette technique permet également d'afficher la valeur précédente quand on revient dans une fonction :

CHX0 : option choisie dans le menu principal.

NR1, NR2, NR6 : numéro de rubrique.

RD1, RD2, RD6 : type de rubrique ("1" pour recettes, "2" pour dépenses).

MA4 : Indicateur de traitement

mensuel ("1") ou annuel ("2").

ANN2, ANN3, ANN4, ANN5, ANN6 : année en traitement.

MOIS2, MOIS4 : mois en traitement.

Matrices

Les matrices sont initialisées avec "OPTION BASE 1".

AN(17) : années existantes (copie de l'enregistrement 0 du fichier des comptes).

Indice : pointeur sur les enregistrements recettes et dépenses (1 à 34) correspondant à l'année (OFF_AN).

MO(2,12) : pointeurs sur les enregistrements de comptes pour une année donnée (copie de 2 enregistrements successifs 1 à 34 du fichier des comptes).

1er indice : type de rubrique (RD?).

2ème indice : mois (MOIS?).

NOM_M\$(12) : noms des mois de l'année.

Indice : numéro du mois (MOIS?).

RUBR\$(2,18,2) : copie du fichier des rubriques.

1er indice : type de rubrique (RN?).

2ème indice : numéro de rubrique (NR?).

3ème indice : nom complet (1) ou abréviation (2).

TOTAL(2) : totalisation des comptes.

Indice : type de rubrique (RD?).

MONT(13,2,19) : contient les montants en cours de traitement.

1er indice : mois (MOIS?). Si = 13, il s'agit d'une totalisation annuelle.

2ème indice : type de rubrique (RD?).

3ème indice : numéro de rubrique (NR?). Si = 19, il s'agit d'une totalisation par rubrique.

□

```
100 !=====
110 != COMPTES FAMILIAUX =
120 !=====
130 !
140 ! (c) J.-Ph.Guillemant
150 !
```

```
160 OPTION BASE 1 :: CALL CLEAR :: ON BR
EAK NEXT :: CALL INIT :: CALL LOAD(-3180
6,16)!interdire touches 'clear' & 'quit'
165 CALL LOAD(8196,62,248):: CALL LOAD(1
6376,77,73,82,69,32,32,160,0):: CALL LOA
```

```
D(-24576,3,0,0,2,2,224,131,224,4,32,0,0)
170 IMAGE ## #####
180 IMAGE ####.##
190 IMAGE #####.##
200 IMAGE #####.##
210 IMAGE #####|#####.##|#####.##|
220 IMAGE #####:#####.##
230 IMAGE REC#####.##IDEP#####.##|
240 IMAGE 1 U.=#####.##
250 !matrices & variables
260 DIM RUBR$(2,18,2),AN(17),MO(2,12),NO
```



```

M_M$(12),MONT(13,2,19),TOTAL(2)
270 FRUB,CHX0,NR1,NR2,ANN2,MOIS2,ANN3,AN
N4,MOIS4,ANN5,NR6,ANN6=0 :: RD1,RD2,MA4,
RD6=1 :: RECN,OFF_AN,I,J,K=0
280 A$="" :: F_RUB$="DSK1.RUBRIQUES" ::
F_MONT$="DSK1.MONTANTS"
290 !noms des mois
300 DATA Janvier,Fevrier,Mars,Avril,Mai,
Juin,Juillet,Aout,Septembre,Octobre,Nove
mbre,Decembre
310 !batons recettes
320 DATA 000000000000007E0000000000007E7
E00000000007E7E7E000000007E7E7E7E
330 DATA 0000007E7E7E7E7E00007E7E7E7E7E7
E007E7E7E7E7E7E7E7E7E7E7E7E7E7E
340 !batons depenses.
350 DATA 7E000000000000007E7E00000000000
07E7E7E00000000007E7E7E7E00000000
360 DATA 7E7E7E7E7E0000007E7E7E7E7E7E000
07E7E7E7E7E7E7E007E7E7E7E7E7E7E7E
370 !batons rubriques
380 DATA 00808080808080800000C0C0C0C0C0C0
000E0E0E0E0E0E00000F0F0F0F0F0F0F000
390 DATA 00F8F8F8F8F8F8F80000FCFCFCFCFCFC
000FEFEFEFEFEFE0000FFFFFFFFFFFFF00
400 !caracteres: 91( ), 92(\), 93( ), 94
(^), 95( ), 124( ), 126(~)
410 CALL CHAR(91,"101010101010107C",92,"
000000FF00FF0000",93,"7C10101010101010",
94,"00008080FF808000")
420 CALL CHAR(95,"00000000FF000000",124,
RPT$("10",8),126,"E0909EE9C9A9090E")
430 ! mise a jour table mois
440 FOR I=1 TO 12 :: READ NOM_M$(I):: NE
XT I
450 ! lecture rubriques
460 ON ERROR 490 :: OPEN #1:F_RUB$,INPUT
,INTERNAL,VARIABLE 2*(19+5)
470 ON ERROR 760 :: FOR I=1 TO 18 :: INP
UT #1:RUBR$(1,I,1),RUBR$(1,I,2),RUBR$(2,
I,1),RUBR$(2,I,2):: NEXT I :: CLOSE #1 :
: FRUB=-1
480 ! chargement anneess
490 ON ERROR 510 :: OPEN #1:F_MONT$,INPU
T,INTERNAL,FIXED 9*18
500 ON ERROR 760 :: FOR I=1 TO 17 :: INP
UT #1:AN(I),:: NEXT I :: INPUT #1:RECN :
: CLOSE #1 :: GOTO 570
510 ON ERROR 760 :: OPEN #1:F_MONT$,OUTP
UT,RELATIVE,INTERNAL,FIXED 9*18
520 RECN=35 :: FOR I=1 TO 17 :: PRINT #1
:AN(I),:: NEXT I :: PRINT #1:RECN
530 FOR I=1 TO 17 :: FOR J=1 TO 2 :: FOR
K=1 TO 11 :: PRINT #1:MO(J,K),:: NEXT K
:: PRINT #1:MO(J,K):: NEXT J :: NEXT I
:: CLOSE #1
540 !
550 ! menu principal
560 !

```

```

570 DISPLAY AT(1,5)ERASE ALL:"COMPTES FA
MILIAUX":TAB(5);RPT$("\",17)
580 RESTORE 640 :: READ I :: FOR J=4 TO
I*2+2 STEP 2 :: READ A$ :: DISPLAY AT(J,
1):A$ :: NEXT J
590 DISPLAY AT(23,1):"QUEL CHOIX DESIREZ
-VOUS?" :: IF CHX0<>0 THEN DISPLAY AT(23
,25):CHX0
600 ACCEPT AT(23,26)VALIDATE(DIGIT)SIZE(
-1)BEEP:A$ :: IF A$="" THEN 600 ELSE CHX
0=VAL(A$):: IF CHX>I THEN 600
610 IF CHX0<>0 AND CHX0<>1 AND CHX0<>I A
ND NOT FRUB THEN DISPLAY AT(24,1)BEEP:"P
AS DE FICHIER RUBRIQUES!!!" :: GOTO 600
620 IF CHX0<>I THEN ON CHX0+1 GOSUB 800,
890,1060,1590,1720,2040,2400 :: GOTO 570
ELSE CALL LINK("MIRE")
630 !menu principal
640 DATA 7
650 ! 0 initialisation fichiers
660 DATA "1 ENTREE DES RUBRIQUES"
670 DATA "2 ENTREE DES COMPTES"
680 DATA "3 ETAT DES COMPTES"
690 DATA "4 BILAN COMPTABLE"
700 DATA "5 EVOLUTION ANNUELLE"
710 DATA "6 EVOLUTION D'UNE RUBRIQUE"
720 DATA "7 FIN DU PROGRAMME"
730 !
740 ! traitement d'erreurs
750 !
760 CALL ERR(I,J,J,J):: DISPLAY AT(24,1)
BEEP:"ERREUR";I;"A LA LIGNE";J :: ACCEPT
AT(24,28)SIZE(1)BEEP:A$ :: CALL CLEAR :
: STOP
770 !
780 ! choix 0: re-init fichiers
790 !
800 DISPLAY AT(24,1):"ETES-VOUS VRAIMENT
SUR ? N" :: ACCEPT AT(24,26)SIZE(-1)BEE
P:A$ :: IF A$<>"0" THEN 850
810 DELETE F_RUB$ :: DELETE F_MONT$ :: F
RUB=0 :: FOR I=1 TO 2 :: FOR J=1 TO 18 :
: FOR K=1 TO 2 :: RUBR$(I,J,K)="" :: NEX
T K :: NEXT J :: NEXT I
820 OPEN #1:F_MONT$,OUTPUT,RELATIVE,INTE
RNAL,FIXED 9*18 :: RECN=35 :: FOR I=1 TO
17 :: AN(I)=0 :: PRINT #1:AN(I),:: NEXT
I :: PRINT #1:RECN
830 FOR I=1 TO 17 :: FOR J=1 TO 2 :: FOR
K=1 TO 11 :: MO(J,K)=0 :: PRINT #1:MO(J
,K),:: NEXT K
840 MO(J,K)=0 :: PRINT #1,MO(J,K):: NEXT
J :: NEXT I :: CLOSE #1
850 RETURN
860 !
870 ! choix 1: modification rubriques
880 !
890 DISPLAY AT(1,4)ERASE ALL:"RECETTES";
TAB(18);"DEPENSES":TAB(4);RPT$("_",8);TA

```

```

B(18);RPT$(" ",8)
900 FOR I=1 TO 18 :: FOR J=1 TO 2 :: IF
RUBR$(J,I,1)="" THEN A$=RPT$(" ",10)ELSE
A$=SEG$(RUBR$(J,I,1),1,10)
910 DISPLAY AT(I+3,(J-1)*14+1):USING 170
:I,A$ :: NEXT J :: NEXT I
920 !modification
930 DISPLAY AT(23,1):"ENTREZ 1 POUR RECE
TTE":TAB(8);"2 POUR DEPENSE:";RD1 :: ACC
EPT AT(24,24)VALIDATE("12")SIZE(-1)BEEP:
RD1
940 DISPLAY AT(23,1):"RUBRIQUE A MODIFIE
R:" :: IF NR1<>0 THEN DISPLAY AT(23,2
1):NR1
950 ACCEPT AT(23,22)VALIDATE(DIGIT)SIZE(
-2)BEEP:A$ :: IF A$="" THEN 950 ELSE NR1
=VAL(A$):: IF NR1<1 OR NR1>18 THEN 950
960 DISPLAY AT(23,1):"ENTREZ LE NOM DE L
A RUBRIQUE":RUBR$(RD1,NR1,1);TAB(22);RUB
R$(RD1,NR1,2):: ACCEPT AT(24,1)SIZE(-18)
BEEP:RUBR$(RD1,NR1,1)
970 DISPLAY AT(NR1+3,(RD1-1)*14+4)SIZE(1
0):RUBR$(RD1,NR1,1):: IF RUBR$(RD1,NR1,1
)="" THEN RUBR$(RD1,NR1,2)="" :: GOTO 99
0
980 DISPLAY AT(23,1):"ABREVIATION (4 CAR
.) ?" :: DISPLAY AT(24,22)SIZE(4):RUBR$(
RD1,NR1,1):: ACCEPT AT(24,22)SIZE(-4)BEE
P:RUBR$(RD1,NR1,2)
990 DISPLAY AT(23,1):"ENCORE (O/N): 0:"
" :: ACCEPT AT(23,15)VALIDATE("ON")SIZE(
-1)BEEP:A$ :: IF A$="0" THEN 930
1000 !écriture disque
1010 OPEN #1:F_RUB$,OUTPUT,INTERNAL,VARI
ABLE 2*(19+5)
1020 FOR I=1 TO 18 :: PRINT #1:RUBR$(1,I
,1),RUBR$(1,I,2),RUBR$(2,1,1),RUBR$(2,I
,2):: NEXT I :: CLOSE #1 :: FRUB=-1 :: RE
TURN
1030 !
1040 ! choix 2: entree des comptes
1050 !
1060 DISPLAY AT(1,5)ERASE ALL:"ENTREE DE
S COMPTES":TAB(5);RPT$(" ",18): "ANNEE
(19XX)": "MOIS (1-12)":
1070 DISPLAY AT(8,1):"RECETTES (1)": "DEP
ENSES (2)":;RD2 :: IF ANN2<>0 THEN DISPL
AY AT(4,14):ANN2
1080 IF MOIS2<>0 THEN DISPLAY AT(6,14):M
OIS2
1090 ACCEPT AT(4,15)VALIDATE(DIGIT)SIZE(
-2)BEEP:A$ :: IF A$="" THEN 1090 ELSE AN
N2=VAL(A$)
1100 ACCEPT AT(6,15)VALIDATE(DIGIT)SIZE(
-2)BEEP:A$ :: IF A$="" THEN 1100 ELSE MO
IS2=VAL(A$):: IF MOIS2<1 OR MOIS2>12 THE
N 1100
1110 ACCEPT AT(9,15)VALIDATE("12")SIZE(-
1)BEEP:RD2

```

```

1120 !recherche montants par type & mois
1130 OFF_AN=0 :: J=1 :: FOR I=1 TO 17 ::
IF AN(1)=ANN2 THEN OFF_AN=I*2-1 ELSE IF
AN(I)<>0 THEN J=J+1
1140 NEXT I :: IF OFF_AN<>0 THEN 1160
1150 IF J=18 THEN DISPLAY AT(24,2)BEEP:"
PAS PLUS DE 17 ANNEES !" :: GOTO 1090 E
LSE OFF_AN=J*2-1 :: AN(J)=ANN2
1160 DISPLAY AT(24,1):" " :: OPEN #1:F_MO
NT$,INTERNAL,RELATIVE,UPDATE,FIXED 9*18
:: RESTORE #1,REC OFF_AN+RD2-1
1170 FOR I=1 TO 11 :: INPUT #1:MO(RD2,I)
,:: NEXT I :: INPUT #1:MO(RD2,I)
1180 FOR I=1 TO 18 :: MONT(MOIS2,RD2,I)=
0 :: NEXT I :: IF MO(RD2,MOIS2)=0 THEN 1
200
1190 RESTORE #1,REC MO(RD2,MOIS2):: FOR
I=1 TO 17 :: INPUT #1:MONT(MOIS2,RD2,I),
:: NEXT I :: INPUT #1:MONT(MOIS2,RD2,I)
1200 DISPLAY AT(11,1):"MODIF. D'UNE RUBR
IQUE: 0" :: ACCEPT AT(11,24)VALIDATE("ON
")SIZE(-1)BEEP:A$ :: IF A$="N" THEN 1380
1210 !modification d'une rubrique
1220 CALL HCHAR(4,1,32,8*32):: DISPLAY A
T(15,1):RPT$(" ",28):: FOR I=1 TO 18 ::
IF RUBR$(RD2,I,1)="" THEN 1240
1230 A$=SEG$(RUBR$(RD2,I,1),1,10):: J=IN
T((I-1)/2):: DISPLAY AT(J+16,(I-1-J*2)*1
4+1):USING 170:I,A$
1240 NEXT I
1250 DISPLAY AT(4,1):"NO DE RUBRIQUE :":
" " :: IF NR2<>0 THEN DISPLAY AT(4,17):NR
2
1260 DISPLAY AT(7,1):"ANCIENNE VALEUR:"
" A AJOUTER...": " A DEDUIRE...":TAB
B(18);RPT$(" ",8):"NOUVELLE VALEUR:"
1270 ACCEPT AT(4,18)VALIDATE(DIGIT)SIZE(
-2)BEEP:A$ :: IF A$="" THEN 1270 ELSE NR
2=VAL(A$):: IF NR2<1 OR NR2>18 THEN 1270
1280 IF RUBR$(RD2,NR2,1)="" THEN 1270 EL
SE DISPLAY AT(4,1):" "
1290 I=LEN(RUBR$(RD2,NR2,1)): DISPLAY A
T(4,14-1/2):RUBR$(RD2,NR2,1):TAB(14-1/2
);RPT$(" ",1)
1300 DISPLAY AT(7,17):USING 190:MONT(MOI
S2,RD2,NR2):: DISPLAY AT(11,17):USING 19
0:MONT(MOIS2,RD2,NR2)
1310 ACCEPT AT(8,18)VALIDATE(NUMERIC)SIZ
E(-9)BEEP:A$ :: IF A$="" THEN 1330 ELSE
DISPLAY AT(8,17):USING 190:VAL(A$)
1320 MONT(MOIS2,RD2,NR2)=MONT(MOIS2,RD2,
NR2)+VAL(A$):: DISPLAY AT(11,17):USING 1
90:MONT(MOIS2,RD2,NR2)
1330 ACCEPT AT(9,18)VALIDATE(NUMERIC)SIZ
E(-9)BEEP:A$ :: IF A$="" THEN 1350 ELSE
DISPLAY AT(9,17):USING 190:VAL(A$)
1340 MONT(MOIS2,RD2,NR2)=MONT(MOIS2,RD2,
NR2)-VAL(A$):: DISPLAY AT(11,17):USING 1
90:MONT(MOIS2,RD2,NR2)

```

```

1350 DISPLAY AT(13,1):"ENCORE (O/N): 0"
:: ACCEPT AT(13,15)VALIDATE("ON")SIZE(-1)BEEP:A$
1360 IF A$="0" THEN DISPLAY AT(13,1):"
:: GOTO 1250 ELSE 1520
1370 !affichage tous montants
1380 TOTAL(RD2)=0 :: IF RD2=1 THEN A$="R
ECETTES " ELSE A$="DEPENSES "
1390 A$=A$&NOM M$(MOIS2)&" 19"&STR$(ANN2)
:: DISPLAY AT(1,(28-LEN(A$))/2)ERASE AL
L:A$:TAB((28-LEN(A$))/2);RPT$("_",LEN(A$)
))
1400 FOR I=1 TO 18 :: IF RUBR$(RD2,I,1)=
" THEN 1430
1410 J=I+5 :: DISPLAY AT(I+3,1):RUBR$(RD
2,I,1);RPT$(".",19-LEN(RUBR$(RD2,I,1)))
1420 IF MONT(MOIS2,RD2,I)<>0 THEN DISPLA
Y AT(I+3,20):USING 190:MONT(MOIS2,RD2,I)
:: TOTAL(RD2)=TOTAL(RD2)+MONT(MOIS2,RD2,
I)
1430 NEXT I :: DISPLAY AT(J-1,20):RPT$("_
",9):: DISPLAY AT(J,19):USING 200:TOTAL
(RD2)
1440 !entree des montants
1450 FOR I=1 TO 18 :: IF RUBR$(RD2,I,1)=
" THEN 1490
1460 TOTAL(RD2)=TOTAL(RD2)-MONT(MOIS2,RD
2,I)
1470 ACCEPT AT(I+3,20)VALIDATE(NUMERIC)S
IZE(-9)BEEP:A$ :: IF A$="" THEN MONT(MOI
S2,RD2,I)=0 ELSE MONT(MOIS2,RD2,I)=VAL(A
$)
1480 DISPLAY AT(I+3,20):USING 190:MONT(M
OIS2,RD2,I):: TOTAL(RD2)=TOTAL(RD2)+MONT
(MOIS2,RD2,I):: DISPLAY AT(J,19):USING 2
00:TOTAL(RD2)
1490 NEXT I :: DISPLAY AT(J,1)SIZE(16):"
CORRECT (O/N): 0"
1500 ACCEPT AT(J,16)VALIDATE("ON")SIZE(-
1)BEEP:A$ :: IF A$<>"0" THEN CALL HCHAR(
J,3,32,16):: GOTO 1450
1510 !mise a jour fichier
1520 IF MO(RD2,MOIS2)=0 THEN MO(RD2,MOIS
2)=RECN :: RECN=RECN+1
1530 RESTORE #1,REC MO(RD2,MOIS2):: FOR
I=1 TO 17 :: PRINT #1:MONT(MOIS2,RD2,I),
:: NEXT I :: PRINT #1:MONT(MOIS2,RD2,I)
1540 RESTORE #1,REC OFF_AN+RD2-1 :: FOR
I=1 TO 11 :: PRINT #1:MO(RD2,I),:: NEXT
I :: PRINT #1:MO(RD2,I)
1550 RESTORE #1 :: FOR I=1 TO 17 :: PRIN
T #1:AN(I),:: NEXT I :: PRINT #1:RECN ::
CLOSE #1 :: RETURN
1560 !
1570 ! choix 3: etat des comptes
1580 !
1590 DISPLAY AT(1,6)ERASE ALL:"ETAT DES
COMPTES":TAB(6);RPT$("\",16): "ANI J F
M A M J J A S O N D I":RPT$("_",28)

```

```

1600 OPEN #1:F_MONT$,INPUT ,RELATIVE,INT
ERNAL,FIXED 9*18 :: ANN3=0
1610 FOR I=1 TO 17 :: K=ANN3 :: ANN3=100
:: OFF_AN=0 :: FOR J=1 TO 17 :: IF AN(J)
>K AND AN(J)<ANN3 THEN ANN3=AN(J):: OFF
_AN=J
1620 NEXT J :: IF OFF_AN=0 THEN DISPLAY
AT(I+5,1):"--I";RPT$(" ",12);"I" :: GOT
O 1670
1630 RESTORE #1,REC OFF_AN*2-1 :: FOR J=
1 TO 2 :: FOR K=1 TO 11 :: INPUT #1:MO(J
,K),:: NEXT K :: INPUT #1:MO(J,K):: NEXT
J
1640 DISPLAY AT(I+5,1):USING 170:ANN3 ::
DISPLAY AT(I+5,3):"I"
1650 FOR J=1 TO 12 :: IF MO(1,J)=0 THEN
IF MO(2,J)=0 THEN A$="--" ELSE A$="D" EL
S IF MO(2,J)=0 THEN A$="R" ELSE A$="~"
1660 DISPLAY AT(I+5,J*2+3):A$ :: NEXT J
:: DISPLAY AT(I+5,28):"I"
1670 NEXT I :: DISPLAY AT(23,1):RPT$("_
",28):: CLOSE #1 :: ACCEPT AT(24,28)SIZE(
1)BEEP:A$ :: IF A$="P" THEN CALL IMP_ECR
1680 RETURN
1690 !
1700 ! choix 4: bilan
1710 !
1720 DISPLAY AT(1,6)ERASE ALL:"BILAN COM
PTABLE":TAB(6);RPT$("\",15): "MENSUEL (
1)": "ANNUEL (2) ":MA4: "ANNEE (19XX):
"
1730 IF ANN4<>0 THEN DISPLAY AT(7,14):AN
N4
1740 ACCEPT AT(5,15)VALIDATE("12")SIZE(-
1)BEEP:MA4 :: IF MA4=2 THEN 1760
1750 DISPLAY AT(9,1):"MOIS (1-12):" ::
IF MOIS4<>0 THEN DISPLAY AT(9,14):MOIS4
1760 ACCEPT AT(7,15)VALIDATE(DIGIT)SIZE(
-2)BEEP:A$ :: IF A$="" THEN 1760 ELSE AN
N4=VAL(A$)
1770 DISPLAY AT(24,1):" " :: IF MA4=2 THE
N 1790
1780 ACCEPT AT(9,15)VALIDATE(DIGIT)SIZE(
-2)BEEP:A$ :: IF A$="" THEN 1780 ELSE MO
IS4=VAL(A$):: IF MOIS4<1 OR MOIS4>12 THE
N 1780
1790 OFF_AN=0 :: FOR I=1 TO 17 :: IF AN(
I)=ANN4 THEN OFF_AN=I*2-1
1800 NEXT I :: IF OFF_AN=0 THEN DISPLAY
AT(24,1):"CETTE ANNEE N'EXISTE PAS !" :
: GOTO 1760
1810 OPEN #1:F_MONT$,INTERNAL,RELATIVE,I
NPUT ,FIXED 9*18
1820 RESTORE #1,REC OFF_AN :: FOR I=1 TO
2 :: FOR J=1 TO 11 :: INPUT #1:MO(I,J),
:: NEXT J :: INPUT #1:MO(I,J):: NEXT I :
: IF MA4=2 THEN 1890
1830 !calcul bilan mensuel
1840 IF MO(1,MOIS4)=0 AND MO(2,MOIS4)=0

```



```

THEN CLOSE #1 :: DISPLAY AT(24,1):"CE MO
IS N'EXISTE PAS !" :: GOTO 1760
1850 FOR I=1 TO 2 :: FOR J=1 TO 18 :: MO
NT(MOIS4,I,J)=0 :: NEXT J :: IF MO(I,MOI
S4)=0 THEN 1870
1860 RESTORE #1,REC MO(I,MOIS4):: FOR J=
1 TO 17 :: INPUT #1:MONT(MOIS4,I,J):: N
EXT J :: INPUT #1:MONT(MOIS4,I,J)
1870 TOTAL(I)=0 :: NEXT I :: A$="BILAN "
&NOM_M$(MOIS4)&" 19"&STR$(ANN4):: K=MOIS
4 :: GOTO 1940
1880 !calcul bilan annuel
1890 FOR I=1 TO 2 :: FOR K=1 TO 18 :: MO
NT(13,I,K)=0 :: NEXT K :: FOR J=1 TO 12
:: FOR K=1 TO 18 :: MONT(J,I,K)=0 :: NEX
T K :: IF MO(I,J)=0 THEN 1920
1900 RESTORE #1,REC MO(I,J):: FOR K=1 TO
17 :: INPUT #1:MONT(J,I,K):: MONT(13,I
,K)=MONT(13,I,K)+MONT(J,I,K):: NEXT K
1910 INPUT #1:MONT(J,I,K):: MONT(13,I,K)
=MONT(13,I,K)+MONT(J,I,K)
1920 NEXT J :: TOTAL(I)=0 :: NEXT I :: A
$="BILAN 19"&STR$(ANN4):: K=13
1930 !affichage
1940 CLOSE #1 :: DISPLAY AT(1,(28-LEN(A$
))/2)ERASE ALL:A$:TAB((28-LEN(A$))/2);RP
T$("_",LEN(A$))
1950 FOR I=1 TO 18 :: CALL HCHAR(I+2,2,1
24):: FOR J=1 TO 2 :: IF RUBR$(J,I,1)=""
THEN 1970
1960 DISPLAY AT(I+2,(J-1)*14+1)SIZE(4):R
UBR$(J,I,2):: DISPLAY AT(I+2,(J-1)*14+5)
:USING 190:MONT(K,J,1):: TOTAL(J)=TOTAL(
J)+MONT(K,J,1)
1970 DISPLAY AT(I+2,J*14):"I" :: NEXT J
:: NEXT I :: CALL HCHAR(21,2,95,29):: CA
LL HCHAR(22,2,124)
1980 DISPLAY AT(22,1):USING 230:TOTAL(1)
,TOTAL(2):: CALL HCHAR(23,2,95,29):: DIS
PLAY AT(24,8):"SOLDE: ";TOTAL(1)-TOTAL(2
)
1990 ACCEPT AT(24,28)SIZE(1)BEEP:A$ :: I
F A$="P" THEN CALL IMP_ECR
2000 RETURN
2010 !
2020 ! choix 5: evolution annuelle
2030 !
2040 RESTORE 320 :: FOR I=128 TO 143 STE
P 4 :: READ A$ :: CALL CHAR(I,A$):: NEXT
I :: CALL COLOR(13,13,1,14,9,1)
2050 DISPLAY AT(1,5)ERASE ALL:"EVOLUTION
ANNUELLE":TAB(5);RPT$("\",18): "ANNEE
(19XX)": :: IF ANN5<>0 THEN DISPLAY AT(4
,14):ANN5
2060 ACCEPT AT(4,15)VALIDATE(DIGIT)SIZE(
-2)BEEP:A$ :: IF A$="" THEN 2060 ELSE AN
N5=VAL(A$)
2070 DISPLAY AT(24,1):"" :: OFF_AN=0 ::
FOR I=1 TO 17 :: IF AN(I)=ANN5 THEN OFF_

```

```

AN=I*2-1
2080 NEXT I :: IF OFF_AN=0 THEN DISPLAY
AT(24,1):"CETTE ANNEE N'EXISTE PAS !" :
: GOTO 2060
2090 OPEN #1:F_MONT$,INTERNAL,RELATIVE,I
NPUT ,FIXED 9*18
2100 RESTORE #1,REC OFF_AN :: FOR I=1 TO
2 :: FOR J=1 TO 11 :: INPUT #1:MO(I,J),
:: NEXT J :: INPUT #1:MO(I,J):: NEXT I
2110 FOR I=1 TO 2 :: TOTAL(I),MONT(13,I,
19)=0 :: FOR J=1 TO 12 :: MONT(J,I,19)=0
:: IF MO(I,J)=0 THEN 2140
2120 RESTORE #1,REC MO(I,J):: FOR K=1 TO
17 :: INPUT #1:MONT(J,I,K):: MONT(J,I,
19)=MONT(J,I,19)+MONT(J,I,K):: NEXT K ::
INPUT #1:MONT(J,I,K)
2130 MONT(J,I,19)=MONT(J,I,19)+MONT(J,I,
K):: TOTAL(I)=MAX(TOTAL(I),MONT(J,I,19))
2140 MONT(13,I,19)=MONT(13,I,19)+MONT(J,
I,19):: NEXT J :: NEXT I :: CLOSE #1 ::
A$="EVOLUTION ANNEE 19"&STR$(ANN5)
2150 DISPLAY AT(1,5)ERASE ALL:A$:TAB(5);
RPT$("_",20): :TAB(5);"I RECETTESI DEP
ENSESI":RPT$("_",27)
2160 !evolution en chiffres
2170 FOR I=1 TO 12 :: DISPLAY AT(I+5,1):
USING 210:SEG$(NOM_M$(I),1,3)&".",MONT(I
,1,19),MONT(I,2,19):: NEXT I
2180 DISPLAY AT(18,1):RPT$("_",27):: DIS
PLAY AT(19,1):USING 210:"TOT.",MONT(13,1
,19),MONT(13,2,19)
2190 DISPLAY AT(20,1):RPT$("_",27):: DIS
PLAY AT(21,1):USING 210:"Moy.",MONT(13,1
,19)/12,MONT(13,2,19)/12
2200 DISPLAY AT(22,1):RPT$("_",27):TAB(5
);"SOLDE:" :: DISPLAY AT(23,11):USING 19
0:MONT(13,1,19)-MONT(13,2,19)
2210 ACCEPT AT(24,28)SIZE(1)BEEP:A$ :: I
F A$="P" THEN CALL IMP_ECR
2220 !evolution en batons
2230 CALL HCHAR(4,1,32,21*32):: A$=" [R
E[C[E[I[T[I[E[IS[ [ " :: FOR I=1 TO 10 :: DI
SPLAY AT(I+2,1):SEG$(A$,I*2-1,2);TAB(26)
;"[ [ " :: NEXT I
2240 DISPLAY AT(13,1):"0IJ_F_M_A_M_J_J_A
_S_O_N_DI_I"
2250 A$=" ]D]E]P]E]N]S]E]S] ]" :: FOR I=
1 TO 10 :: DISPLAY AT(I+13,1):SEG$(A$,I*
2-1,2);TAB(26);"] ]" :: NEXT I
2260 FOR I=1 TO 12 :: J=10*8*MONT(I,1,19
)/MAX(TOTAL(1),TOTAL(2)):: K=INT(J/8)::
IF K<>0 THEN CALL VCHAR(13-K,I*2+3,135,K
)
2270 IF INT(J-8*K)<>0 THEN CALL VCHAR(12
-K,I*2+3,128+INT(J-8*K))
2280 J=10*8*MONT(I,2,19)/MAX(TOTAL(1),TO
TAL(2)):: K=INT(J/8):: IF K<>0 THEN CALL
VCHAR(14,I*2+3,143,K)
2290 IF INT(J-8*K)<>0 THEN CALL VCHAR(14

```

```

+K,I*2+3,136+INT(J-8*K))
2300 NEXT I :: J=10*8*MONT(13,1,19)/(12*
MAX(TOTAL(1),TOTAL(2))):: K=INT(J/8):: I
F K<>0 THEN CALL VCHAR(13-K,29,135,K)
2310 IF INT(J-8*K)<>0 THEN CALL VCHAR(12
-K,29,128+INT(J-8*K))
2320 J=10*8*MONT(13,2,19)/(12*MAX(TOTAL(
1),TOTAL(2))):: K=INT(J/8):: IF K<>0 THE
N CALL VCHAR(14,29,143,K)
2330 IF INT(J-8*K)<>0 THEN CALL VCHAR(14
+K,29,136+INT(J-8*K))
2340 DISPLAY AT(24,1):USING 240:MAX(TOTA
L(1),TOTAL(2))/10
2350 ACCEPT AT(24,28)SIZE(1)BEEP:A$ :: I
F A$="P" THEN CALL IMP_ECR
2360 RETURN
2370 !
2380 ! choix 6: evolution d'une rubrique
2390 !
2400 RESTORE 380 :: FOR I=128 TO 135 STE
P 4 :: READ A$ :: CALL CHAR(I,A$):: NEXT
I
2410 DISPLAY AT(1,2)ERASE ALL:"EVOLUTION
D'UNE RUBRIQUE":TAB(2);RPT$("\",24): :
RECETTE (1):"DEPENSE (2)  ":RD6: "AN
NEE (19XX)  ":
2420 IF ANN6<>0 THEN DISPLAY AT(7,16):AN
N6
2430 ACCEPT AT(5,17)VALIDATE("12")SIZE(-
1)BEEP:RD6 :: IF RD6=1 THEN CALL COLOR(1
3,13,1)ELSE CALL COLOR(13,9,1)
2440 ACCEPT AT(7,17)VALIDATE(DIGIT)SIZE(
-2)BEEP:A$ :: IF A$="" THEN 2440 ELSE AN
N6=VAL(A$)
2450 DISPLAY AT(24,1):"" :: OFF_AN=0 ::
FOR I=1 TO 17 :: IF AN(I)=ANN6 THEN OFF_
AN=I*2-1
2460 NEXT I :: IF OFF_AN=0 THEN DISPLAY
AT(24,1):"CETTE ANNEE N'EXISTE PAS !!" :
: GOTO 2440
2470 DISPLAY AT(15,1):RPT$("_",28):: FOR
I=1 TO 18 :: IF RUBR$(RD6,I,1)="" THEN
2490
2480 A$=SEG$(RUBR$(RD6,I,1),1,10):: J=IN
T((I-1)/2):: DISPLAY AT(J+16,(I-1-J*2)*1
4+1):USING 170:I,A$
2490 NEXT I :: DISPLAY AT(9,1):"NO DE RU
BRIQUE:" :: IF NR6<>0 THEN DISPLAY AT(9,
16):NR6
2500 ACCEPT AT(9,17)VALIDATE(DIGIT)SIZE(
-2)BEEP:A$
2510 IF A$="" THEN 2500 ELSE NR6=VAL(A$)
:: IF NR6<1 OR NR6>18 THEN 2500 ELSE IF
RUBR$(RD6,NR6,1)="" THEN 2500
2520 OPEN #1:F_MONT$,INTERNAL,RELATIVE,I
NPUT ,FIXED 9*18
2530 RESTORE #1,REC OFF_AN+RD6-1 :: FOR
I=1 TO 11 :: INPUT #1:MO(RD6,I):: NEXT
I :: INPUT #1:MO(RD6,I)
2540 TOTAL(RD6),MONT(13,RD6,NR6)=0 :: FO

```

```

R 1=1 TO 12 :: MONT(1,RD6,NR6)=0 :: IF M
O(RD6,I)=0 THEN 2580
2550 RESTORE #1,REC MO(RD6,I):: IF NR6=1
THEN 2570
2560 FOR J=1 TO NR6-1 :: INPUT #1:MONT(1
,RD6,J):: NEXT J
2570 INPUT #1:MONT(1,RD6,NR6):: MONT(13,
RD6,NR6)=MONT(13,RD6,NR6)+MONT(1,RD6,NR6
):: TOTAL(RD6)=MAX(TOTAL(RD6),MONT(1,RD6
,NR6))
2580 NEXT I :: CLOSE #1 :: A$=RUBR$(RD6,
NR6,1)
2590 !affichage en chiffres
2600 DISPLAY AT(1,(28-LEN(A$))/2)ERASE A
LL:A$:TAB((28-LEN(A$))/2);RPT$("_",LEN(A
$))
2610 DISPLAY AT(3,9):"ANNEE 19"&STR$(ANN
6):TAB(9);RPT$("_",10): :TAB(5);RPT$("_",
20)
2620 FOR I=1 TO 12 :: DISPLAY AT(I+7,5):
USING 220:NOM_M$(I),MONT(1,RD6,NR6):: NE
XT I :: DISPLAY AT(20,5):RPT$("_",20)
2630 DISPLAY AT(21,5):USING 220:"TOTAL",
MONT(13,RD6,NR6):: DISPLAY AT(22,5):RPT$
("_",20)
2640 DISPLAY AT(23,5):USING 220:"Moyenne
",MONT(13,RD6,NR6)/12 :: ACCEPT AT(24,28
)SIZE(1)BEEP:A$ :: IF A$="P" THEN CALL I
MP_ECR
2650 !affichage en batons
2660 CALL HCHAR(7,1,32,18*32):: DISPLAY
AT(7,1):RPT$("_",5);RPT$("",20)
2670 FOR I=1 TO 12 :: DISPLAY AT(I+7,1):
SEG$(NOM_M$(I),1,3);".1" :: IF MONT(1,RD
6,NR6)=0 THEN 2700
2680 J=20*8*MONT(1,RD6,NR6)/TOTAL(RD6)::
K=INT(J/8):: IF K<>0 THEN CALL HCHAR(I+
7,8,135,K)
2690 IF INT(J-8*K)<>0 THEN CALL HCHAR(I+
7,8+K,128+INT(J-8*K))
2700 NEXT I :: DISPLAY AT(20,1):RPT$("_",
5);RPT$("",20):: DISPLAY AT(21,1):"moy
.1" :: IF MONT(13,RD6,NR6)=0 THEN 2730
2710 J=20*8*MONT(13,RD6,NR6)/(12*TOTAL(R
D6)):: K=INT(J/8):: IF K<>0 THEN CALL HC
HAR(21,8,135,K)
2720 IF INT(J-8*K)<>0 THEN CALL HCHAR(21
,8+K,128+INT(J-8*K))
2730 DISPLAY AT(22,1):RPT$("_",5);RPT$("
",20):: DISPLAY AT(24,1):USING 240:TOTA
L(RD6)/20
2740 ACCEPT AT(24,28)SIZE(1)BEEP:A$ :: I
F A$="P" THEN CALL IMP_ECR
2750 RETURN
2760 !
2770 !impression ecran
2780 !
2790 SUB IMP_ECR
2800 DISPLAY AT(24,1):"FONCTION NON REAL
ISEE" :: ACCEPT AT(24,28)SIZE(1)BEEP:A$
2810 SUBEND

```

Mission spéciale

Edouard Krok

N.D.L.R. : ce jeu d'aventure est écrit en Basic Etendu.

Des diamants ont été dérobés par l'infâme prince Krapul, qui les a cachés dans son château perdu au fin fond du désert ! Personne ne sait où il se trouve exactement et les rares audacieux qui ont osé partir à sa recherche n'en sont jamais revenus. Serez-vous volontaire pour cette mission bien spéciale qui consiste à retrouver et à ramener les diamants volés ?

Si vous avez un goût prononcé pour l'aventure, la découverte et l'imprévu, stoppez là votre lecture et partez immédiatement à la recherche du château.

Si vous préférez vous faire une idée des petites misères qui vous attendent, quelques explications vous seront peut-être utiles.

Au début, vous suivez un petit chemin de campagne et vous décidez de la direction à prendre en appuyant sur l'une des quatre touches fléchées. Votre premier objectif sera de trouver le passage qui mène au désert et à ses pistes peu visibles où votre sens de l'orientation sera mis à rude épreuve. Dès que vous aurez repéré le château, n'hésitez pas à y pénétrer : il est peu probable que vous ayez besoin d'explorer les 25 salles qui le composent pour trouver les diamants. A ce moment, votre but sera de repartir en ville.

Pour cela, vous pourrez tout naturellement refaire en sens inverse le chemin qui vous a conduit au château en retraversant le désert, puis la campagne pour revenir à votre point exact de départ où il vous sera possible d'entrer dans la ville, si vous avez au moins un diamant sur vous. Ce retour vous sera grandement facilité si vous avez eu la sagesse de faire un plan des zones que vous avez visitées. Une autre solution, plutôt inattendue, consiste à utiliser les souterrains du château qui, si vous ne vous y égarez pas, vous conduiront directement en ville où il vous faudra encore retrouver

vos objets. En espérant que vous n'aurez pas perdu quelques diamants en cours de route, car il ne vous resterait plus qu'à repartir !

Mais n'allez tout de même pas vous imaginer que le prince Krapul va se laisser tranquillement dépouiller : ses acolytes, de monstrueuses créatures qui crachent des flammes mortelles, vont essayer de vous décourager et, au besoin, de vous détruire. Pour avoir une chance de réussir, vous devez donc être armé ; c'est pour cette raison que vous recevrez, avant de partir, une grenade et un pistolet. D'ailleurs, si au cours de votre mission vous perdiez tous vos objets, vous pourriez essayer de revenir à votre point de départ où vous percevriez ces deux armes. Rien ne vous empêche d'ailleurs de laisser les objets que vous transportez dans la campagne avoisinante et de les reprendre au passage.

Si vous rencontrez un ennemi, vous pouvez l'éliminer radicalement avec votre grenade ou tenter de vous défendre avec votre pistolet que vous déplacez de droite à gauche avec les deux touches fléchées "D" et "S".

Le tir se fait en appuyant sur la touche "I" mais, comme il ne s'agit pas d'un pistolet mitrailleur, vous serez obligé de relâcher la gachette (la touche "I") avant de tirer à nouveau. Ces créatures ont la peau dure et il faut souvent les toucher à plusieurs reprises avant qu'elles ne cèdent. Le fantôme, quant à lui, lorsqu'il est atteint, se transforme en une créature différente qui vous attaque aussitôt. Si vous remportez le combat, l'ennemi disparaît définitivement de cet endroit. Si vous perdez, il reste en place, votre pistolet est détruit mais vous pouvez tout de même poursuivre votre route. Si vous n'avez plus de pistolet et que vous rencontrez un ennemi, vous pourrez passer à condition de lui offrir un objet qu'il se choisira lui-même parmi ceux que vous transportez. Ce serait un grand malheur si vous n'aviez plus rien à lui présenter car, dans ce cas, vous n'auriez plus aucun espoir de rentrer un jour chez vous !

Fort heureusement, vous trouverez au hasard de votre voyage des objets qui s'avèreront très utiles. Vous pourrez les prendre, les laisser, voire les déplacer. Si vous avez une clochette, vous l'agiterez chaque fois que vous rencontrerez un serpent : il ira vite se terrer dans son trou et n'en ressortira que lorsque vous vous serez éloigné. Il vous arrivera de trouver un coffre ; vous ne pourrez le prendre car il est trop lourd à porter mais ce sera un jeu d'enfant de l'ouvrir si vous avez la clé. La plupart des coffres renferment un diamant mais, s'il peuvent être vides où contenir un objet utile, un ennemi peut aussi s'y cacher.

Si vous avez de la chance, vous trouverez une carte ; son nom est usurpé car il n'existe en fait aucun plan de ces zones inexplorées, mais une lettre repérant le lieu où vous vous trouvez s'inscrira en haut de l'écran et, si vous le demandez, vous obtiendrez la lettre qui repère l'endroit par où vous êtes entré et celui par lequel vous devez sortir. Contrairement aux premières apparences, la carte est très utile pour se guider dès que l'on s'en sert régulièrement. Les jumelles, quant à elles, vous feront gagner beaucoup de temps car elles permettent de regarder dans toutes les directions sans prendre le risque d'aller voir sur place.

Si vous êtes totalement égaré, que vous vous retrouviez seul, sans arme et sans moral, il vous reste encore la faculté d'abandonner. Si vous tapez "REDO" (FCTN-8) vous pourrez repartir en mission avec un parcours exactement identique, les objets et les ennemis vous attendant aux mêmes endroits. Ainsi, si vous avez fait un plan sur papier, il pourra vous resservir. Si vous tapez "BACK" (FCTN-9) vous partirez chaque fois dans l'inconnu le plus total.

Avant le départ, vous recevez votre ordre de mission puis vous devez vous engager à ramener entre 1 et 5 diamants ; cela conditionne la difficulté de votre tentative. Avec un peu d'expérience, vous constaterez que les problèmes qui se posent pour apporter les 5 diamants à la fois, ne sont pas insurmontables.

□


```

100 ! *****
110 ! * MISSION SPECIALE *
120 ! *****
130 ! * basic etendu *
140 ! *****
150 !
160 DISPLAY AT(7,7)ERASE ALL:"mi
ssion speciale" :: CALL MAGNIFY(
4):: D=1 :: B=2
:: F=8
170 DIM P$(4),A(3),N(3)
180 DEF T(X)=ASC(SEG$(P$(P),X,D)
)
190 CALL CHAR(33,"00FF7060600000
00382472FAF8F8F870FE82AA92AA82FE
FE")
200 CALL CHAR(36,"0000C0C6FFC000
0010101038387CFE106666FFE7E7E77E
E738FE7C38381")
210 FOR X=64 TO 88 STEP F :: CAL
L CHAR(X,"FFFEFCF8F0E0C080FF7F3F
1F0F070301"&RPT$
("F",16)&"0")
220 CALL CHAR(X+4,"FFFFFFFFF3F0
F03FF3F0F03000000000FFFFFFFFFCF
0C0FFFCF0C"):: N
EXT X
230 CALL CHAR(90,"0000000000030F
3F0000000000C0F0FC",68,"02020202
060600000",70,"4
0404040606000000")
240 CALL CHAR(56,"80808000008080
800101010000010101",132,"0101"&R
PT$("0",45))
250 CALL CHAR(128,"F0F0E0E0C0C08
080FFFFFEFEFCFCF8F80F0F070703030
101FFFF7F7F3F3F1
F1F")
260 CALL CHAR(124,"0102020604050
8080911111108081122008040406020A
0909018088888844
422")
270 CALL DELSPRITE(ALL):: CALL C
OLOR(D,5,D,4,B,D):: IF K=15 THEN
RANDOMIZE ELSE
RANDOMIZE 12
280 DISPLAY AT(B,D)ERASE ALL:"vo
tre mission : "
290 DISPLAY AT(7,5):"vous devez
recuperer et": "ramener chez vo
us en ville": "
les diamants derobes par": "l'i
nfame prince Krapul ; "
300 X$="IMLILKONJJCGNKFIMONHCGGG
FIEMELCMNIFIFKGLJIONCFCGFIELILC
LJKNINCOFJKMGLCG
GEFILHILCGGONIMLKFKNKOLCFCGFHIEL
HJCMFJKEOMFKMGOLCGEGF"
310 Q$="222AAABBCDEFHHHIIJJKLL"

```

```

320 FOR P=C TO 4 :: X=INT(25*RND
)+D :: Y=INT(25*RND)+D :: IF X=Y
THEN 320 ELSE P
$(P)=SEG$(X$,INT(4*RND)*25+D,25)
&CHR$(X)&CHR$(Y)
330 FOR X=D TO 25 :: P$(P)=P$(P)
&SEG$(Q$,INT(23*RND)+D,D):: NEXT
X :: NEXT P
340 DISPLAY AT(17,5)BEEP:"combie
n de diamants": "vous sentez vo
us capable": "d'
e ramener ": " : :TAB(8);"choisir
1 a 5"
350 CALL KEY(3,K,S):: IF K<49 OR
K>53 THEN 350 ELSE CALL SOUND(9
9,999,D):: DISPL
AY AT(22,7)ERASE ALL:"preparez v
ous": "vous allez a la campagne
"
360 Q$="2222" :: Q=K-48 :: FOR X
=C TO Q :: CALL MAJ(P$(B),INT(25
*RND)+28,71):: N
EXT X
370 FOR P=C TO 4 :: FOR Y=D TO B
:: CALL MAJ(P$(P),T(25+Y)+27,64
):: NEXT Y :: NE
XT P :: DISPLAY AT(D,16):"1 2 3
4 5" :: CALL HCHAR(D,F,39,Q):: P
,W=C
380 CALL DELSPRITE(ALL):: GOSUB
1530 :: P=P+W :: E=T(26-(W<C))+6
4 :: Q$=Q$&"2" :
: IF P=3 THEN 390 ELSE IF P=B TH
EN GOSUB 1290 ELSE GOSUB 1010
390 FOR X=C TO 3 :: A(X)=D :: N(
X)=C :: NEXT X :: GOTO 410
400 GOSUB 1530 :: W=T(E)-64 :: F
OR X=3 TO C STEP -D :: A(X)=N(X)
:: N(X)=-(W>B)*X
):: W=W-N(X)*B*X :: NEXT X
410 FOR K=-D TO B :: S=N(K+D)::
IF K<B THEN IF S=A(K+D)THEN 440
ELSE 430 ELSE IF
P=3 THEN GOSUB 1460 ELSE IF P=B
THEN GOSUB 1340 ELSE GOSUB 1250
420 GOTO 440
430 IF P=B AND K=C THEN GOSUB 13
20 ELSE IF P=B THEN GOSUB 1360 E
LSE IF P=3 THEN
GOSUB 1400 ELSE GOSUB 1070
440 NEXT K :: IF E>64 THEN E=E-6
4 :: GOTO 400 ELSE Z=T(E+27)-64
450 IF Z<F THEN CALL CHARPAT(32+
Z,X$):: CALL CHAR(60,X$&RPT$("0"
,64))ELSE GOSUB
1490
460 CALL SCREEN(ASC(SEG$("NJPHD"
,P+D,D))-64):: IF POS(Q$,"C",D)T
HEN CALL HCHAR(D

```

```
,14,96+E)
470 ON P+D GOSUB 1540,1560,1570,
1580,1590
480 CALL SPRITE(#4,60,5-3*(P=B)*
(Z>F),129+24*(Z>F),116):: IF LEN
(Q$)>5 THEN Q$=S
EG$(Q$,D,5):: IF Z>F THEN 660 EL
SE 540 ELSE IF Z>F THEN 660
490 W=(E=T(26))-(E=T(27)):: ON W
+B GOTO 510,540,500
500 IF P=4 THEN CALL SCREEN(13):
: CALL TEX(5):: IF Q>C THEN DISP
LAY AT(24,B):"al
lez chercher les diamants" :: GO
TO 550 ELSE 960 ELSE 520
510 IF P=C THEN IF Q$="00000" TH
EN CALL MAJ(Q$,D,65):: CALL MAJ(
Q$,B,66):: GOTO
540 ELSE IF POS(Q$,"G",D)=C THEN
540 ELSE W=4
520 CALL TEX(P+W)
530 CALL KEY(C,K,S):: IF K=79 TH
EN CALL HCHAR(22,D,32,15):: DISP
LAY AT(22,10):"v
ous allez" :: GOTO 380 ELSE IF K
<>78 THEN 530
540 IF Z>C AND Z<9 THEN X$=" un
objet" :: DISPLAY AT(21,D)BEEP:"
":" p prend
re";X$=" la5 utiliser";X$="1 1
a5 laisser";X$ ELSE CALL TEX(7)
550 CALL KEY(C,K,S):: IF S THEN
CALL SOUND(99,999,C)ELSE 550
560 IF K>48 AND K<54 THEN 840
570 IF K=80 THEN X=POS(Q$,"0",D)
:: IF X=C OR Z>7 THEN 650 ELSE C
ALL DELSPRITE(#4
):: CALL MAJ(Q$,X,Z+64):: CALL M
AJ(P$(P),E+27,64):: Q=Q+(Z=7)::
Z=C :: GOTO 540
ELSE IF K<>76 OR Z>F THEN 600
580 CALL KEY(C,K,S):: IF K>48 AN
D K<54 THEN Z=ASC(SEG$(Q$,K-48,D
))-64 ELSE 580
590 IF Z THEN CALL MAJ(Q$,K-48,6
4):: CALL MAJ(P$(P),E+27,Z+64)::
Q=Q-(Z=7):: GOT
O 450 ELSE 650
600 IF K=68*N(C)THEN E=E+D :: GO
TO 400
610 IF K=69*N(D)THEN E=E-5 :: GO
TO 400
620 IF K=83*N(B)THEN E=E-D :: GO
TO 400
630 IF K=88*N(3)THEN E=E+5 :: GO
TO 400
640 IF K=15 OR K=6 THEN 270
650 FOR X=15 TO C STEP -D :: CAL
L SOUND(400-25*X,150+90*X,X):: N
```

```
EXT X :: Z=INT(4
*RND+9):: GOTO 450
660 FOR X=C TO 250 STEP 50 :: CA
LL SOUND(400,400-X,6,800+3*X,3):
: CALL SOUND(500
,111,30):: NEXT X :: X=POS(Q$,"A
",D):: IF X THEN 690 ELSE IF Q$=
"00000" THEN 970
670 Y=INT(5*RND)+D :: X$=SEG$(Q$
,Y,D):: IF X$="G" THEN Q=Q+D
680 IF X$="0" THEN 670 ELSE CALL
MAJ(Q$,Y,64):: GOTO 490
690 CALL MAJ(Q$,X,64):: CALL SPR
ITE(#3,68,B,149,90+76*INT(B*RND)
,#D,124,9,D,256,
#B,132,B,D,256,-25,C):: FOR X=C
TO 1000*RND
700 CALL KEY(C,K,S):: IF S THEN
IF K>48 AND K<54 THEN Y=ASC(SEG$
(Q$,K-48,D))-64
:: IF Y=B OR Y=5 THEN 850 ELSE 7
20 ELSE 720
710 NEXT X
720 ON ERROR 770
730 CALL KEY(C,K,S):: W=K*S=49 :
: CALL POSITION(#4,X,X,#3,Y,Y)::
S=(ABS(X-Y)<12)
*(W=C)
740 IF W-S=C THEN CALL MOTION(#3
,C,26*(K=68)*(Y<165)-26*(K=83)*(
Y>70),#4,C,(Z-F)
*9*SGN(Y-X)):: CALL LOCATE(#D,D,
256,#B,D,256):: GOTO 730
750 CALL POSITION(#4,X,X,#3,Y,Y)
:: CALL LOCATE(#B,145,256+(256-Y
)*W,#D,125,256-(
256-X)*S,#9+5*W,105,4*2*RND+X-Z-
Z)
760 CALL SOUND(-90,-S-5,C):: CAL
L COINC(#4,#B,23,W):: CALL COINC
(#3,#D,24,S):: O
N D-W-W-S GOTO 730,780,800
770 RETURN 720
780 CALL SOUND(300,-D,C):: CALL
MOTION(#3,C,C,#4,C,C):: CALL DEL
SPRITE(#B,#D)::
CALL COLOR(#3,16)
790 FOR X=C TO 220 :: CALL SCREE
N(10-6*(X>20)):: NEXT X :: GOTO
830
800 CALL SOUND(300,-D,C):: CALL
MOTION(#4,C,C,#3,C,C):: IF Z=10
THEN Z=INT(4*RND
)+9 :: GOSUB 1490 :: GOTO 720
810 FOR X=D TO 12 :: CALL COLOR(
#4,9,#B,D):: CALL SOUND(10,-F,X/
B):: CALL COLOR(
#4,1):: CALL SOUND(10,-F,X/B)::
NEXT X :: Z=C
```

```

820 CALL POSITION(#3,X,Y):: W=PO
S(Q$, "2", D):: IF X*W THEN CALL M
AJ(Q$, W, 65)
830 CALL DELSPRITE(#3, #B):: CALL
MAJ(P$(P), E+27, 64+Z):: GOTO 490
840 Y=ASC(SEG$(Q$, K-48, D))-64
850 ON Y+D GOTO 650, 650, 860, 870,
880, 890, 910, 950
860 CALL SOUND(300, -7, C):: CALL
MAJ(Q$, K-48, 64):: GOTO 810
870 CALL TEX(6):: FOR X=C TO D :
: CALL HCHAR(22+B*X, 21, T(26+X)+9
6):: NEXT X :: G
OTO 550
880 IF Z<>F THEN 650 ELSE X=INT(
26*RND):: Z=X+(X-7)*(X>12):: CAL
L SOUND(200, 500,
D):: CALL SOUND(150, 900, D):: CAL
L MAJ(P$(P), 27+E, 64+Z):: GOTO 45
0
890 FOR X=-D TO 3*RND :: CALL SO
UND(400, 784, C):: CALL SOUND(400,
585, 3):: CALL SO
UND(800, 784, 3):: NEXT X
900 IF Z=9 THEN CALL DELSPRITE(#
4):: GOTO 820 ELSE IF Z>9 THEN 7
20 ELSE 550
910 CALL TEX(9):: FOR X=C TO 3 :
: X$="rien" :: IF N(X) THEN W=E-(
X=C)+(X=B)+5*(X=
D)-5*(X=3) ELSE 940
920 IF W=T(26) OR W=T(27) THEN X$=
"sortie" :: GOTO 940 ELSE Y=T(27
+W)-64
930 IF Y THEN IF Y<F THEN X$=CHR
$(32+Y) ELSE IF Y=F THEN X$="coff
re" ELSE X$="enn
emi"
940 DISPLAY AT(21+X-B*(X=C), 20-5
*X-10*(X=3)-LEN(X$)/B) BEEP SIZE(
6):X$ :: NEXT X
:: GOTO 550
950 CALL MAJ(Q$, K-48, 64):: Q=Q+D
:: GOTO 550
960 CALL SCREEN(B):: DISPLAY AT(
23, 7): "mission accomplie" :: FOR
X=D TO 4 :: FOR
Y=150*X TO 500*X STEP 50 :: CAL
L SOUND(60, Y, C):: NEXT Y :: NEXT
X :: CALL SCREE
N(13):: GOTO 980
970 CALL TEX(F):: FOR X=22 TO D
STEP -D :: CALL SOUND(500-X*X, 11
0+12*X, X/B):: NE
XT X
980 FOR X=C TO 990 :: NEXT X
990 CALL SOUND(40, 2000, 9):: DISP
LAY AT(24, 4): "tapez `redo` ou `
back`" :: FOR X=

```

```

C TO 30 :: CALL KEY(C, K, S):: IF
K=15 OR K=6 THEN CALL SOUND(200,
800, B):: GOTO 27
0
1000 NEXT X :: DISPLAY AT(24, D):
" " :: GOTO 990
1010 FOR X=C TO F :: CALL HCHAR(
3+X, 3, 82, 28):: CALL HCHAR(12+X, 3
, 66, 28):: NEXT X
:: CALL HCHAR(12, 3, 74, 28)
1020 IF P<4 THEN 1040 ELSE CALL
CHAR(40, "00010313333300003333130
3010000008080808
08080000080808080808")
1030 CALL CHAR(44, "10181C1C1C1C0
0001C1C1C1C1810000000000080C0C00
000C0C08"):: RET
URN
1040 IF P THEN CALL CHAR(40, "010
A25174F3F5FBF5F3F4F17250A010000A
048D0E4F8F4FAF4F
8E4D048A"):: CALL SPRITE(#5, 40, D
, 25, 80):: RETURN
1050 CALL CHAR(40, "21777FFFFF3F3
E0C0", 42, "88D8FEFFFFFEFE640", 44,
"81C67F070E18000
00", 46, "8040E0C", 47, "")
1060 CALL SPRITE(#5, 40, D, 36, D, C,
-4, #6, 40, D, 26, D, C, -B, #7, 40, D, 23,
10, C, -B, #F, 44, D,
30, D, C, B):: RETURN
1070 Z=12-(K=-D)-9*K :: Y=ABS(K)
:: IF S-Y THEN 1130 ELSE CALL HC
HAR(20, 13-10*K, 6
7, F*S):: CALL HCHAR(20, 16.5-5.5*
K, 129-K, S)
1080 FOR X=16 TO 19 :: CALL HCHA
R(X, 11-F*K, 31, 12*S):: CALL HCHAR
(X-4, Z, 74, 10-S):
: NEXT X
1090 CALL HCHAR(15, Z, 67, 10-S)::
FOR X=13-1.5*P TO -14*(P<>D):: C
ALL HCHAR(X, Z, 66
, 10-S):: NEXT X
1100 IF P<4 THEN CALL HCHAR(11, Z
, 82, -F*(K=C)):: RETURN ELSE IF K
=C THEN CALL DEL
SPRITE(#5, #7)
1110 FOR X=3 TO 6 :: CALL HCHAR(
X, Z, 82, 10-S):: NEXT X
1120 FOR Y=B TO 6 STEP 4 :: FOR
X=C TO D :: CALL VCHAR(9, Z+Y+X+(
K=D), 74, 6):: NEX
T X :: NEXT Y :: CALL HCHAR(11, Z
, 66, 9):: RETURN
1130 FOR X=13+10*(P=4) TO 20*Y ::
CALL HCHAR(X, Z, 66, 9+S):: NEXT X
:: FOR X=12 TO
15*S :: CALL HCHAR(X, Z, 66, 10)::

```



```

NEXT X
1140 FOR Z=K+(K=C)TO K-(K=C)STEP
B :: W=Z=D :: FOR X=3*Y TO B+6*
Y+6*(P=D):: CALL
HCHAR(12+X,15-X*Z,67,4):: CALL
HCHAR(12+X,17+W-B*Z-Z*X,65+W)::
NEXT X
1150 FOR X=Y*(B-N(D)/B)TO 1.5+2.
5*Y STEP .5 :: CALL HCHAR(12+B*X
,18+3.5*W-X*Z,12
9+X-INT(X)-Z):: NEXT X
1160 IF P=4 THEN 1190 ELSE IF P
THEN CALL HCHAR(16,21+9*W,66,D-S
):: GOTO 1240
1170 FOR X=D*Y TO .5+4.5*Y STEP
.5 :: CALL HCHAR(11.5+X,20+W*7-B
*X*Z,77.5+Z-X+IN
T(X)):: CALL HCHAR(11.5+X,3-(19+
B*INT(X))*(Z=-D),74,9-B*INT(X)-
(X=5)):: NEXT X
1180 CALL HCHAR(11,16.5-2.5*Z,90
.5-.5*Z,S):: GOTO 1240
1190 FOR X=3*Y TO B+4*Y :: CALL
HCHAR(9-X,19+5*W-X*Z,80-W):: NEX
T X :: FOR X=D T
O 4+3*S :: CALL HCHAR(B+X,F+X,82
,18-B*X):: NEXT X
1200 IF S THEN CALL HCHAR(10,14,
66,6):: CALL HCHAR(10,15,82,4)::
CALL HCHAR(11,1
5,82,4):: CALL SPRITE(#6+Z,42+Z+
Z,D,68,116-Z*27):: GOTO 1240
1210 FOR X=C TO B :: CALL VCHAR(
3,29+25*W+X*Z,74,18):: NEXT X ::
CALL HCHAR(3,16
.5-10.5*Z,88-W):: CALL HCHAR(20,
16.5-10.5*Z,73+W)
1220 FOR X=D TO B :: C=17+W-8*Z+
X*Z :: CALL VCHAR(7,C,74,10):: C
ALL HCHAR(5+X,C,
88-W):: CALL HCHAR(18-X,C,73+W):
: NEXT X :: C=0
1230 CALL HCHAR(11,23+19*W,66,7)
1240 NEXT Z :: RETURN
1250 FOR X=4-4*N(D)TO F :: CALL
HCHAR(12+X,15.5-X/B,31,B+X+X/B-I
NT(X/B)):: NEXT
X :: CALL HCHAR(12,18,130,N(D)):
: CALL HCHAR(20,22,130,S)
1260 CALL HCHAR(20,11,128,S):: C
ALL HCHAR(20,11,67,12-12*S):: CA
LL HCHAR(15,12,6
7):: CALL HCHAR(15,21,67):: IF P
=4 THEN 1280 ELSE IF P THEN RETU
RN
1270 FOR X=-D TO D STEP B :: CAL
L VCHAR(13,16.5+(4.5+N(D))*X,56.
5+X/2-N(D)*X,-B*
(N(D)=N(D-X)):: NEXT X :: RETUR

```

```

N
1280 CALL VCHAR(7,22,56,-F*(N(C)
=N(D))):: CALL VCHAR(7,11,57,-F*
(N(D)=N(B))):: R
ETURN
1290 FOR X=3 TO 20 :: CALL HCHAR
(X,3,74,28):: NEXT X :: FOR X=D
TO 5 :: X$=RPT$(
"B",4+4*X):: DISPLAY AT(F-X,11-X
*B)SIZE(F+X*4):"J\"&X$&"^_"
1300 DISPLAY AT(15+X,11-X*B)SIZE
(F+X*4):"NO"&X$&"ML" :: NEXT X :
: CALL HCHAR(17,
14,67,6):: CALL HCHAR(18,13,67,F
)
1310 FOR X=C TO 3 :: CALL CHAR(4
0+X,RPT$("0",13-4*X)&RPT$("F",3+
4*X)):: CALL CHA
R(44+X,RPT$("0",4*X)&"F0"&RPT$("
F",14-4*X)):: NEXT X :: RETURN
1320 FOR X=F TO -15*(S=C):: CALL
HCHAR(X,13,31,F):: NEXT X :: DI
SPLAY AT(16,12)S
IZE(6):"0CCCCA" :: CALL HCHAR(16
,14,66,-6*(S=C))
1330 FOR X=15 TO 18*S :: CALL VC
HAR(10,X,82,6):: NEXT X :: RETUR
N
1340 FOR X=19 TO -20*(S=C):: CAL
L HCHAR(X,10,66,14):: NEXT X
1350 FOR X=D TO B*S :: X$=RPT$("
C",F+B*X):: DISPLAY AT(18+X,10-X
)SIZE(10+B*X):"2
"&X$&"A" :: NEXT X :: RETURN
1360 W=K=D :: FOR X=C TO D :: FO
R Y=C TO D :: Z=17+W-6*K-(B*X+Y)
*K :: CALL VCHAR
(9,Z,74+S*F,9):: CALL HCHAR(17+X
,Z,78+K+F*S-Y):: NEXT Y
1370 FOR Y=D TO 3+X :: CALL HCHA
R(17+X,17+W-(B*X+Y)*K,66+S):: NE
XT Y :: CALL HCH
AR(17+X,17+W-(3+X)*K,65+W,-(S=C)
):: NEXT X
1380 IF S THEN IF W THEN DISPLAY
AT(9,5)SIZE(4):",-./" ELSE DISP
LAY AT(9,21)SIZE
(4):"()*+*"
1390 RETURN
1400 Z=12-(K=-D)-9*K :: Y=S+ABS(
K)=D :: FOR X=C TO F-(K=0):: CAL
L VCHAR(3,Z+X,74
-Y,14+4*ABS(K)):: NEXT X :: IF Y
THEN 1420
1410 FOR X=6 TO F :: CALL HCHAR(
X-F*(X=F),Z,66,9-(K=C)):: CALL H
CHAR(X+11,Z+(K=-
D),67,10*ABS(K)):: NEXT X :: CAL
L HCHAR(20,30+(K=D)*27,75,S):: R

```

```

ETURN
1420 W=N(D)=C :: FOR X=-3*W TO 5
:: CALL HCHAR(F-X,15-X,74,4+B*X)
:: NEXT X :: Y=
ABS(K):: FOR Z=K+(K=C)TO K-(K=C)
STEP B
1430 FOR X=3*Y TO B+3*Y :: W=19+
(Z=D)*5-Z*X :: CALL HCHAR(F-X,W,
72-(Z=D)):: CALL
HCHAR(15+X,W,65+(Z=D)):: NEXT X
1440 FOR X=1.5*Y TO 1+4.5*Y STEP
.5 :: W=19+(Z=D)*5-B*X*Z :: CAL
L HCHAR(8.5-X,W,
87+(Z=D)*B+(X>INT(X)):: CALL VC
HAR(9.5-X,W,74,5+B*X)
1450 CALL HCHAR(14+X,W,76-(Z=D)*
B-(X>INT(X)):: NEXT X :: NEXT Z
:: FOR X=15 TO
18*S :: CALL VCHAR(9,X,31,6):: N
EXT X :: RETURN
1460 CALL HCHAR(20,4,66,26):: FO
R X=B-B*N(D)TO 4+S :: CALL HCHAR
(15+X,15-X,67,4+
B*X):: NEXT X
1470 CALL HCHAR(20,9,64,S):: CAL
L HCHAR(20,24,65,S):: CALL HCHAR
(17,21,65+B*N(C)
):: CALL HCHAR(17,12,64+3*N(B))
1480 CALL VCHAR(F,11,57,F+F*(N(D)
)+N(B)=D):: CALL VCHAR(F,22,56,
F+F*(N(C)+N(D)=D
):: RETURN
1490 DATA 3F7F7700777F7F7F7F0000
0000000000FCFEEEE0EEFEFEFEFE
1500 DATA 020207070A0F0702020203
01000F3FFF0000000080800000000000
80C0F0FCFF,03078
5C7673F3F1F0F0F07070303010180C04
0C1C7FEFCF8F8F0F0E0E0C0C08
1510 DATA 03050702011F3F7FDF8F87
C3860C08388040C08000F0F8FCF6E2C2
87C2602038,030E1
02405030507020F1B31F0000000E0380
424A2C2A2E448F8D88C0F
1520 RESTORE :: FOR X=F TO Z ::
READ X$ :: NEXT X :: CALL CHAR(6
0,X$):: RETURN
1530 CALL HCHAR(D,14,32):: Y=INT
(15*RND)+B :: CALL COLOR(C,Y,Y,B
,Y,Y,13,Y,Y):: F
OR X=4 TO F :: CALL COLOR(X,D):
: CALL COLOR(X,Y,Y):: NEXT X ::
RETURN
1540 CALL COLOR(C,15,15,4,B,13,5
,13,12,6,4,13,7,F,D,F,13,F,13,12
,15)

```

```

1550 CALL COLOR(#5,16,#6,15,#7,1
5,#F,9):: RETURN
1560 CALL COLOR(C,11,11,5,12,12,
6,12,D,7,F,D,13,12,11):: CALL CO
LOR(#5,12):: RET
URN
1570 CALL COLOR(C,6,6,B,F,5,5,14
,12,6,5,14,7,F,12,F,14,5):: RETU
RN
1580 CALL COLOR(C,B,B,4,12,7,5,1
1,12,6,7,11,7,11,7):: RETURN
1590 CALL COLOR(C,15,15,4,B,10,5
,10,14,6,12,14,7,F,10,F,10,12,13
,14,15):: CALL C
OLOR(#5,12,#7,12):: RETURN
1600 SUB TEX(A)
1610 CALL HCHAR(21,1,32,128):: I
F A=9 THEN SUBEXIT ELSE IF A<5 T
HEN DISPLAY AT(2
2,1)BEEP:"o n desirez vous alle
r"
1620 ON A+1 GOTO 1630,1640,1650,
1660,1670,1680,1690,1700,1710
1630 DISPLAY AT(24,9):"a la camp
agne" :: SUBEXIT
1640 DISPLAY AT(24,6):"affronter
le desert" :: SUBEXIT
1650 DISPLAY AT(24,6):"visiter l
e chateau" :: SUBEXIT
1660 DISPLAY AT(24,5):"dans les
souterrains" :: SUBEXIT
1670 DISPLAY AT(24,11):"en ville
" :: SUBEXIT
1680 DISPLAY AT(22,2)BEEP:"vous
etes arrive chez vous" :: SUBEXI
T
1690 DISPLAY AT(22,11):"entree":
:TAB(11);"sortie" :: SUBEXIT
1700 DISPLAY AT(21,1)BEEP:"depla
cement":TAB(9);"par touches flec
hees": "tir par t
ouche 1": "abandon `redo` ou `ba
ck`" :: SUBEXIT
1710 DISPLAY AT(22,2):"krapul a
eu raison de vous": "ici sera
votre tombe"
1720 SUBEND
1730 SUB MAJ(A$,R,C)
1740 A$=SEG$(A$,1,R-1)&CHR$(C)&S
EG$(A$,R+1,99):: IF LEN(A$)>6 TH
EN SUBEXIT
1750 IF C=64 THEN CALL SOUND(500
,250,5)ELSE CALL SOUND(300,900,4
)
1760 CALL HCHAR(1,17+2*R,C-32)
1770 SUBEND

```

Crayon optique

Bernard Bazuel

Ce programme, écrit en assembleur, permet de créer des dessins directement à l'écran à l'aide d'un crayon optique. Cet accessoire (qu'il vous faudra construire en suivant les indications jointes à cet article) n'est pas indispensable car le programme peut aussi recevoir des ordres provenant du clavier.

Les fichiers source qui apparaissent pages suivantes concernent une version utilisable avec une extension de mémoire 32Ko. Cependant, nos lecteurs munis seulement de la Mini-mémoire pourront utiliser ce programme en entrant, sous EASY BUG, la liste de codes machines placée en fin d'article (une autre solution, plus simple, consiste à charger le code objet situé sur la cassette d'accompagnement du numéro 8, avec l'option L de EASY BUG; le programme est baptisé "CRAYON"). La différence entre les deux versions porte essentiellement sur l'affichage des menus, qui ont dû être raccourcis afin de permettre leur implantation dans les 4Ko de la Mini-mémoire. Pour la version "éditeur/assembleur", nous avons prévu d'autres fonctions sous forme de modules qui seront ajoutés par la suite; cela ne sera bien sûr pas possible avec la version "Mini-mémoire".

Nous ne nous étendrons pas sur le fonctionnement du programme car les nombreuses remarques présentes dans les listings vous permettront de comprendre aisément. De même, il n'est pas nécessaire de donner un mode d'emploi pour la version "éditeur/assembleur", les divers menus étant très "parlants". Ce n'est pas tout à fait aussi simple avec la version "Mini-mémoire"; nous vous proposons donc un petit mode d'emploi dans lequel les chiffres représentent les touches qu'il faut presser pour accéder à une fonction; les lettres minuscules entre parenthèses correspondent aux mêmes fonctions, mais cette fois avec le crayon optique.

Menu principal

- 1 - (p) : point
- 2 - (t) : trait
- 3 - (d) : droite
- 4 - (c) : curviligne
- 5 - (e) : écran

Options

1 : Point

Mise en place d'un point

2 : Trait

- 1 (v) visible
le déplacement de la mire laisse une trace visible (pixels allumés).
- 2 (i) invisible
efface les points visibles (pixels éteints).
- 3 (c) couleur
choix d'une nouvelle couleur (code des 16 couleurs de 0 à F).
- 4 (f) fond
l'écran est colorisé en couleur de fond (pixels éteints).
- 5 (r) remplissage
la surface intérieure à la ligne fermée est colorisée (pixels allumés) avec choix de couleur code 0 à F).

3 : Droite

- 1 (a) axes
trace deux axes perpendiculaires (pixels allumés).
- 2 (s) segment
place une mire fixe en extrémité d'un segment (1er choix), puis trace un segment jusqu'à la mire mobile (2ème choix).
- 3 (h) horizontale
trace une droite horizontale.
- 4 (v) verticale
trace une droite verticale.
- (t)
retour au menu principal.

4 : curviligne

remplissage de surface autour d'un point pivot.

- 1 (a)
en allumant les pixels.
- 2 (e)
en coloriant le fond (pixels éteints).

4 (s)

confirmation ou annulation de l'option remplissage curviligne.

5 (p)

mise en place du point pivot avec choix de la couleur (code 0 à F).

(t)

retour au menu principal.

5 : écran

nettoyage de l'écran (pixels éteints).

Les options *visible*, *invisible*, *fond* et *curviligne* s'annulent en les choisissant deux fois. La mire se déplace au crayon optique ou avec les touches *W*, *E*, *R*, *S*, *D*, *Z*, *X* et *C*.

Liste des composants nécessaires

- 1 Photo-transistor Texas type TIL81
- 2 amplis opérationnels LF156 (circuits intégrés, boîtier 8 broches)
- 4 résistances 82K Ω , 1/4W
- 1 résistance 68K Ω , 1/4W
- 1 résistance 2,2K Ω , 1/4W
- 1 condensateur 47nF, 250V
- 1 condensateur 82nF, 250V
- 1 connecteur CANON 9 broches femelle avec capot
- 1,5 mètre de câble 4 conducteurs blindés
- 1 plaquette d'essai à bandes, de 1 cm * 10 cm (4 rangées de 39 trous). Les bandes cuivrées doivent être perpendiculaires à la grande longueur de la plaquette.
- 1 tube plastique de 10mm de diamètre intérieur et de 16 cm de longueur (glisser la plaquette cablée à l'intérieur du tube, montage légèrement serré)
- 2 embouts plastiques, un côté cellule photo transistor, un côté sortie câble vers console. Fermer le tout en collant au trichlo
- 2 portes fusibles à boulonner
- 2 fusibles de 300mA

Les connexions sur la plaquette sont à faire en fil souple, fin et isolés (type téléphone), environ 2 mètres.

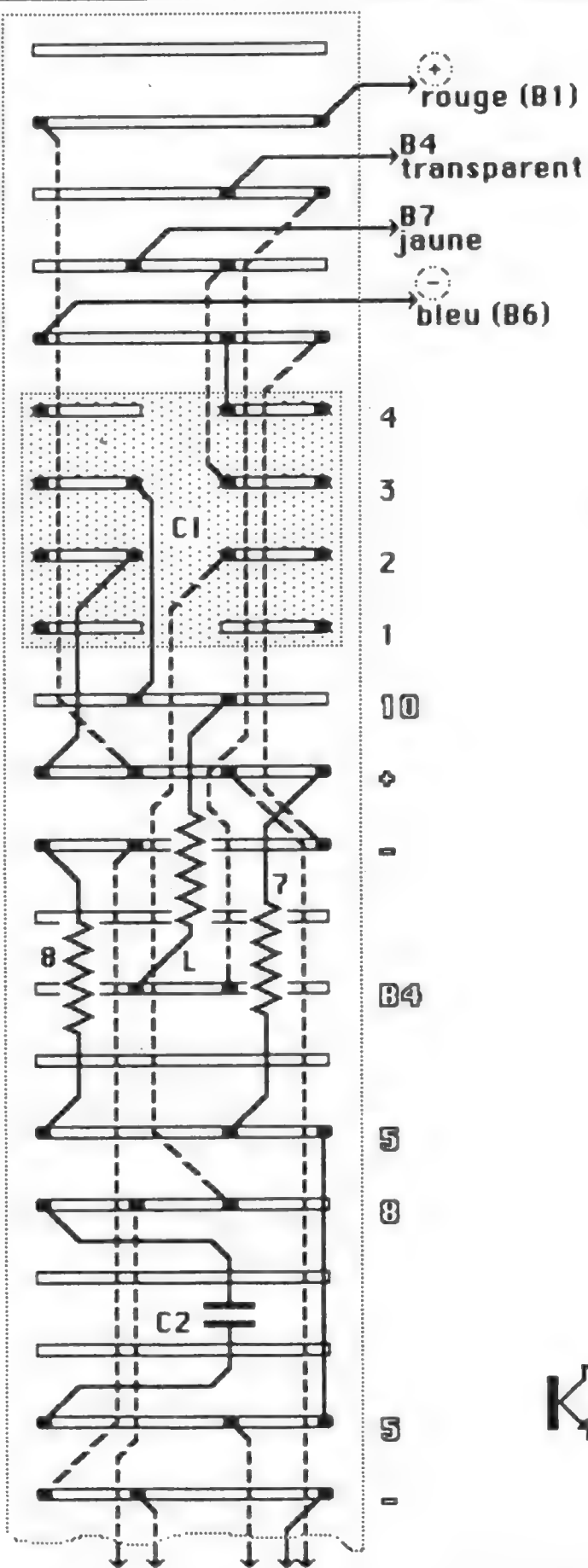


Schéma d'implantation

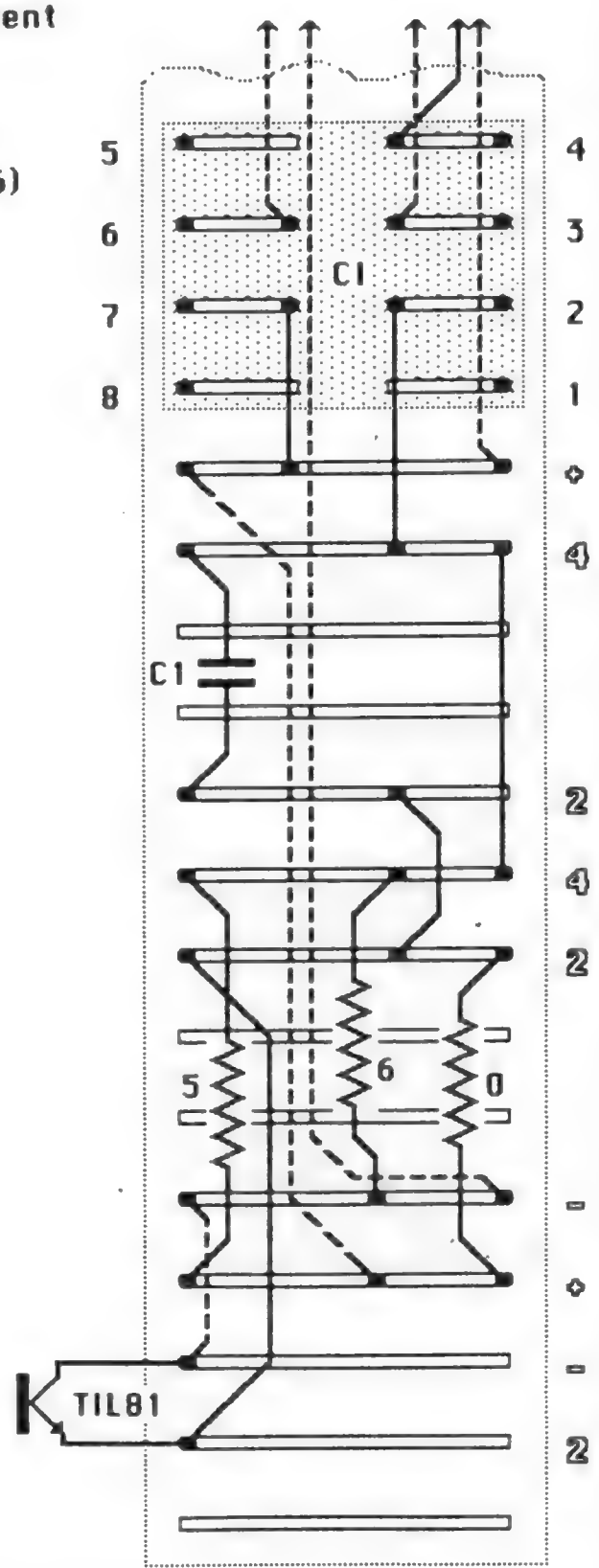
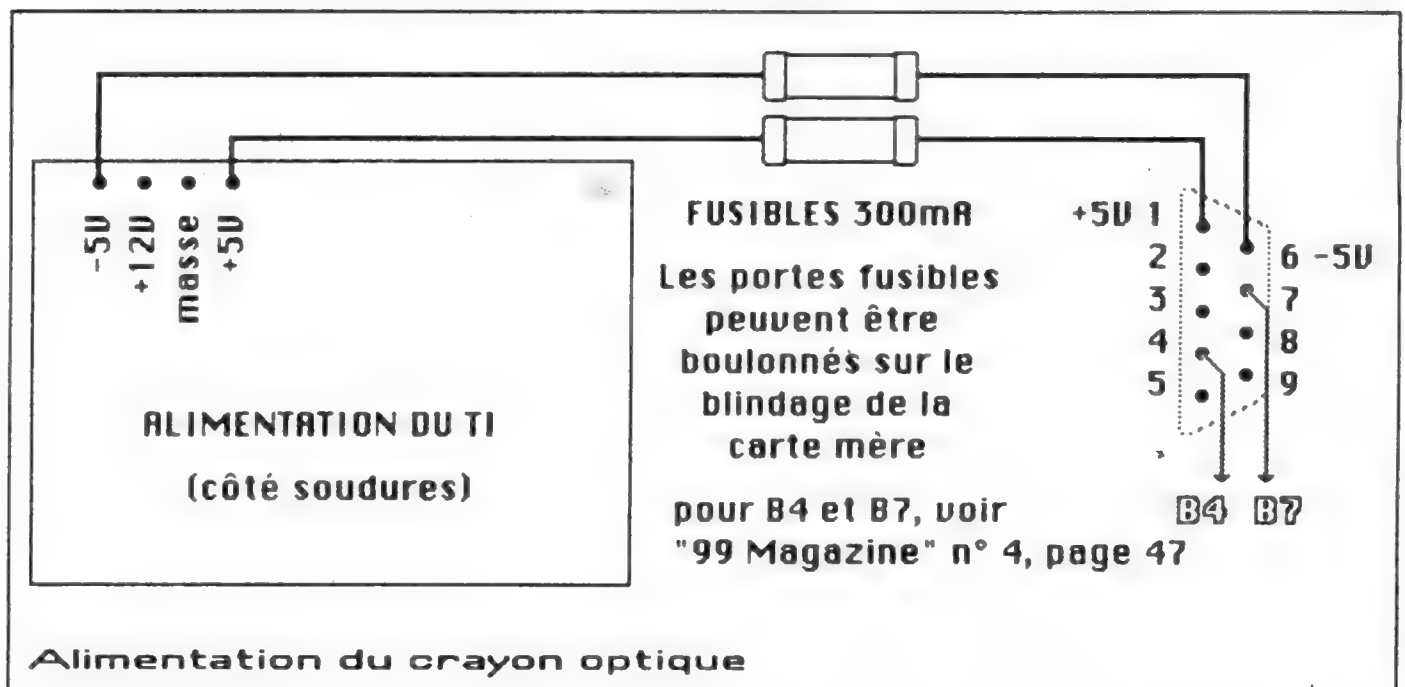
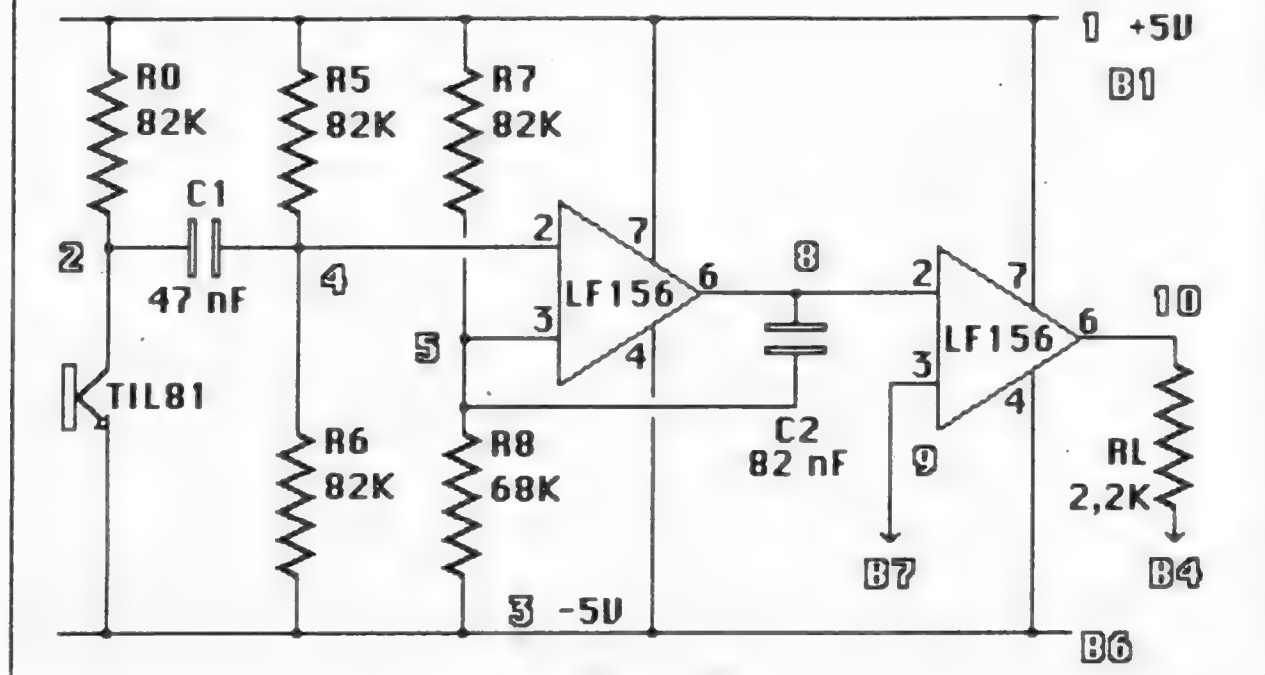


Schéma théorique



Source COPY/S

```
COPY "DSK1.REFE/S"
COPY "DSK1.REMP/S"
COPY "DSK1.CRAY/S"
COPY "DSK1.STOR/S"
COPY "DSK1.MENU/S"
COPY "DSK1.NEWC/S"
COPY "DSK1.GRAF/S"
COPY "DSK1.PROG/S"
```

Source REFE/S

```
*****
*
* PROGRAMME DE DESSIN A L'ECRAN
*
* AVEC LE CRAYON OPTIQUE
*
* OU LE CLAVIER
*
*****
*
* VERSION EDITOR/ASSEMBLER
*
*****
*
* COPYRIGHT 99 MAGAZINE
*
* &
*
* BERNARD BEZUEL
*
*****
*
* PROGRAMME DE LANCEMENT:CRAYON
*
* EXECUTION AVEC L'OPTION BASIC
*
* DISQUETTE DANS LECTEUR DSK1
*
*****
```

```
*
** REFE/S ** REFERENCES/DATA **
DEF CRAYON
REF USBJ,UMBJ,USBR,UMBR
REF DSRLNK,KSCAN,WAIR
AORG >2700
PAB EQU >1F00
PNTR EQU >8356
RTCLAV EQU >8375
STATUS EQU >837C
TABLUT EQU >1B80
USRWS EQU >20BA
*CARACTERES MINUSCULES NOIR SUR BLANC (CODE 32 A 95)
TALPHA DATA >0000,>0000,>0000,>0000,>0000 espace
DATA >FEF,>EFEF,>EFF,>EFF
DATA >FD7,>D7D7,>FFFF,>FFFF
DATA >FFF,>D783,>D783,>D7FF
DATA >FEF,>C3AF,>C7EB,>87FF
DATA >FF9F,>9BF7,>EFD9,>89FF
DATA >FDF,>AFAF,>DFAB,>87CB
DATA >FFF,>FBFB,>F7FF,>FFF
DATA >FFF,>EFD,>DFDF,>EFF
DATA >FFF,>F7FB,>FBFB,>F7FF
DATA >FFF,>D7EF,>83EF,>D7FF
DATA >FFF,>EFEF,>83EF,>EFF
DATA >FFF,>FFF,>CFCF,>EFD
DATA >FFF,>FFF,>87FF,>FFF
DATA >FFF,>FFF,>FCCF,>CFFF
DATA >FFF,>FBF7,>EFD,>BFFF
DATA >FE7,>D8DB,>D8DB,>E7FF
DATA >FE7,>D7F7,>F7F7,>E3FF
DATA >FE7,>D8FB,>F7EF,>C3FF
DATA >FE3,>FBF7,>FBDB,>E7FF
DATA >FF7,>E7D7,>D7C3,>F7FF
DATA >FE3,>EFE7,>FBDB,>E7FF
DATA >FE7,>DFC7,>D8DB,>E7FF
DATA >FCC3,>FBF7,>EFEF,>EFF
DATA >FE7,>D8E7,>D8DB,>E7FF
DATA >FE7,>D8E3,>FBDB,>E7FF
DATA >FFF,>FEF,>FFF,>EFF
DATA >FFF,>FEF,>FCCF,>EFD;
```

TAMPON ASCII (512 ADRESSES MEMOIRE)
BLOCK POUR ACCES DISQUE
POINTEUR POUR FICHIER SUR DISQUE
RETOUR CLAVIER
BYTE D'ETAT
TABLE D'ATTRIBUTS DES LUTINS
WORSPEACE UTILISATEUR

DATA >FFFF,>F7EF,>DDEF,>F7FF <
 DATA >FFFF,>FFFF,>C3FF,>C3FF =
 DATA >FFFF,>DDEF,>F7EF,>DFFF >
 DATA >FFCF,>B7EF,>EFFF,>EFFF ?
 DATA >FFFF,>E308,>D30F,>E3FF ?
 DATA >FFFF,>C787,>B787,>C8FF a
 DATA >9FDF,>C7D8,>D8D8,>87FF b
 DATA >FFFF,>C788,>BFB8,>C7FF c
 DATA >F3F7,>C787,>B787,>C3FF d
 DATA >FFFF,>C788,>83BF,>C7FF e
 DATA >E7D8,>DF8F,>DFDF,>DFFF f
 DATA >FFE3,>D8D8,>E3FB,>D8E7 g
 DATA >9FDF,>D7C8,>D8D8,>D8FF h
 DATA >EFFF,>8FEF,>EFEF,>83FF i
 DATA >F7FF,>E7F7,>F7F7,>87CF j
 DATA >DFDF,>D8D7,>CFD7,>D8FF k
 DATA >CFEF,>EFEF,>EFEF,>83FF l
 DATA >FFFF,>57AB,>ABAB,>ABFF m
 DATA >FFFF,>A7D8,>D8D8,>D8FF n
 DATA >FFFF,>C788,>8888,>C7FF o
 DATA >FFFF,>87D8,>D8C7,>DF8F p
 DATA >FFFF,>C387,>B7C7,>F7E3 q
 DATA >FFFF,>A7D8,>DFDF,>DFFF r
 DATA >FFFF,>C78F,>C7FB,>87FF s
 DATA >DFDF,>87DF,>DFD8,>E7FF t
 DATA >FFFF,>B787,>B787,>CFFF u
 DATA >FFFF,>8888,>D7D7,>EFFF v
 DATA >FFFF,>95D5,>D5D5,>E8FF w
 DATA >FFFF,>B8D7,>EFD7,>B8FF x
 DATA >FFFF,>D8D8,>D8E3,>F8C7 y
 DATA >FFFF,>8387,>EFD8,>83FF z
 DATA >FFFF,>CFDF,>DFDF,>CFFF [
 DATA >FFFF,>FFDF,>E7F7,>F8FF \
 DATA >FFFF,>E7F7,>F7F7,>E7FF]
 DATA >FEFF,>D788,>FFFF,>FFFF .
 DATA >FFFF,>FFFF,>FFFF,>C3FF _

*BUFFER/DATA DIVERS

GRAFWS BSS >20 *ADRESSE.>2900*

SVCLAV DATA 0

SVY DATA 0

SVX DATA 0

ESPACE DE TRAVAIL ROUTINES GRAPHIQUES

SAUVEGARDE R0 (RETOUR CLAVIER)

SAUVEGARDE Y

SAUVEGARDE X

SVCOL DATA 0
 COLORS DATA >2000
 TABCOL DATA >2100,>3100,>4100,>5100
 DATA >6100,>7100,>8100,>9100
 DATA >A100,>B100,>C100,>D100
 DATA >E100,>F100,>0100,>1100
 MENU DATA MENU10
 MASQUE DATA MENU00
 MENU00 BYTE 32,32,32,32,32,32
 BYTE 32,32,32,32,32,32
 BYTE 32,32,32,32,32,32
 BYTE 32,32,32,32,32,32
 BYTE 32,32,32,32,32,32
 XMENU BYTE >08,>38,>68,>98,>C8,>F8
 YMENU BYTE >B7
 YLUTIN BYTE >AE,>AA,>A5,>A5,>A5,>AE,>CO
 OUTECR DATA >C000
 SENSXY DATA >2220,>2101,>1110,>1202
 LIGNES DATA 0,0
 COORD DATA >AE18,>0000
 ATRIBU DATA >C01C,>000F
 DATA >C01C,>010F
 DATA >C01C,>020F
 DATA >C018,>030F
 DATA >C013,>040F
 DATA >C013,>050F
 DATA >C013,>060F
 DATA >C018,>070F
 DATA >C018,>0A01
 DATA >C000,>080F
 DATA >AA18,>0808
 DATA >AA18,>090F
 DATA >C008,>080F
 DATA >C010,>080F
 DATA >C018,>080F
 DATA >C020,>080F,>D000
 LUTINS DATA >2030,>7C7F,>7F3F,>3F1F
 DATA >030F,>3F3F,>1F0F,>0300
 DATA >1F3F,>3F7F,>7FFC,>3000
 DATA >FEFC,>7C78,>3830,>0000
 DATA >F8FC,>FCFE,>FE3E,>0C04
 SAUVEGARDE COULEUR (MENU)
 COULEUR INITIALE
 PALETTE COULEURS
 PREMIER MENU AFFICHE
 LIGNE 24 VIDE
 ABSCSSE LUTINS MASQUE OPTION
 ORDONNEE LUTINS MASQUE OPTION
 ORDONNEE POUR EFFACEMENT LUTIN
 DIRECTION DE DEPLACEMENT SUR L'ECRAN
 2 POINTS POUR SEGMENT
 COORDONNEES DU POINT SUR ECRAN
 LUTIN 1
 1 2
 3 4
 5 6
 7 8
 MASQUE
 MIRE SEGMENT
 MIRE
 FLASH
 MENU ALLUME
 MENU ALLUME
 MENU ALLUME
 MENU ALLUME
 LUTIN 1 CODE >0
 1
 2
 3
 4
 5
 4

```

**** REMPLISSAGE SURFACE ***
RTREMP DATA 0 *
RTABOUT DATA 0 *
RTILIN DATA 0 * SAUVEGARDE RETOUR APPEL (R11)
RTSECT DATA 0 *
RTBIT DATA 0 *
HB DATA 0 SENS REMPLISSAGE VERTICAL (-1 HAUT, 1 BAS)
GGD DATA 0 SENS REMPLISSAGE HORIZONTAL (-1 GAUCHE, 1 DROITE)
DAX DATA 0 ABSCISSE EXTREME DROITE AVANT-DERNIERE LIGNE
GAV DATA 0 ABSCISSE EXTREME GAUCHE AVANT-DERNIERE LIGNE
GAP DATA 0 ABSCISSE EXTREME GAUCHE DERNIERE LIGNE
AR DATA 0,0 *
DATA 0,0 *
DATA 0,0 *
DATA 0,0 *
DATA 0,0 *
DATA 0,0 *
DATA 0,0 *
DATA 0,0 *

```

```
DATA 0,0 * DONNEES DES POINTS D'ARRET  
DATA 0,0 * Y/X,HB (X=POINT DROIT DU SEGMENT)  
  
DATA 0,0 *  
DATA 0,0 *  
DATA 0,0 *  
DATA 0,0 *  
DATA 0,0 *  
DATA 0,0 *  
DATA 0,0 *  
DATA 0,0 *  
REPLI MOV R11,RTREMP  
LWPI GRAFWS  
MOV ZCOORD,ZAR Y/X DANS ARRO  
AB ZP1+1,ZAR Y+1  
MOV ZP1,ZAR+2 HB-1  
LI R3,4 INCREMENT STOCKAGE POINTS D'ARRET  
AI R3,-4 DEBUT DU REMPLISSAGE D'UN SECTEUR  
JLT RE150 PLUS DE SECTEUR A REMPLIR  
CLR R0  
MOVB ZAR+1(R3),R0 ABSCISSE BORNE DROITE  
SWPB R0  
CLR R1  
MOVB ZAR(R3),R1 ORDONNEE BORNE DROITE  
SWPB R1  
MOV ZAR+2(R3),ZHB SENS DU REMPLISSAGE VERTICAL  
MOV R0,ZGAV EXTREMITE GAUCHE SEGMENT PRECEDENT  
MOV R0,ZDAV EXTREMITE DROITE SEGMENT PRECEDENT  
*CONTROLE MIRE SUR POINT DEJA ALLUME  
MOV R0,R10  
BL GREBIT  
JEQ RE150  
BL GREILIN TRACE D'UN SEGMENT  
CI R1,4 CONTROLE ECRAN HAUT  
JLT RE105  
CI R1,180 CONTROLE ECRAN BAS  
JGT RE105  
MOV ZM1,ZGD CONTROLE CONTINUITE A GAUCHE AVEC SEE  
C ZGAV,ZGAP MEME EXTREMITE GAUCHE  
JEQ RE120  
C ZGAV,ZGAP  
JGT RE115 115-NOUVELLE EXTREMITE A GAUCHE DE  
MOV ZGAP,R10 NOUVELLE EXTREMITE A DROITE DE L'AND
```

MOV 2GAV,R12	R10:POINT DE RECHERCHE;R12: BUTEE DE RECHERCHE	JEQ RE140	140:CONTINUER LA RECHERCHE
DEC R12		RE145 C	R10,2GAV BORNE A GAUCHE EXTREME GAUCHE SECTEUR PRECEDENT ?
BL 2RESECT	NOUVEAUX SECTEURS A CREER ?	JLT RE105	SI OUI:SECTEUR FERME->PASSER AU SECTEUR SUIVANT
JMP RE120	POUR RECHERCHE A DROITE	MOV R10,2DAV	DAV:NOUVELLE BORNE DROITE POUR SEGMENT SUIVANT
RE115 MOV 2GAV,R10	NOUVELLE EXTREME A GAUCHE DE L'ANCIENNE	JMP RE110	FIN DES CALCULS,REMPLIR LE SEGMENT SUIVANT
MOV 2GAP,R12	R10:POINT DE RECHERCHE;R12: BUTEE DE RECHERCHE	RE150 LWP1 USRWS	FIN DU PROGRAMME
NEG 2HB	REMPILSSAGE EN SENS INVERSE POUR NOUVEAUX SECTEURS	MOV 2RTREMP,R11	
A 2HB,R1	RETOUR POUR RECHERCHE SUR SEGMENT PRECEDENT	B *R11	
DEC R12		*** SOUS-PROGRAMMES ***	
BL 2RESECT	NOUVEAUX SECTEURS A CREER ?	*TRACE D'UN SEGMENT HORIZONTAL ENTRE DAV ET PIXEL ALLUME A GAUCHE DE DAV	
NEG 2HB	REPRENDRE ANCIEN SENS DE REMPLISSAGE	REILIN MOV R11,2RTILIN	
A 2HB,R1	RETOUR SUR DERNIER SEGMENT	MOV 2DAV,R0	BORNE DROITE
RE120 MOV 2GAP,2GAV	FIN DU CONTROLE A GAUCHE->NOUVELLE BORNE GAUCHE GAV	MOV R0,R2	
A 2HB,R1	CALCULS BORNE,CONTINUITE DROITE AVEC SEGME. PRECED.	MOV R0,R10	
MOV 2DAV,R10	R10:POINT DE RECHERCHE	BL 2REBIT	ALLUMER DANS R4 BIT CORRESPONDANT A DAV
BL 2REBIT	PIXEL ALLUME ?	BL 2EXTREM	REMPILSSAGE EXTREME DROITE (BITS A GAUCHE DE DAV)
JEQ RE135	135:PIXEL ALLUME,CHERCHER BORNE VERS LA GAUCHE	CI R4,20100	OCTET-CORRESPONDANT A R7 PLEIN A GAUCHE ?
MOV 2P1,260	DEBUT RECHERCHE BORNE VERS LA DROITE	JNE RE165	165:BORNE GAUCHE ATTEINTE,PASSER AU SEGMENT SUIVANT
LI R12,253	BORNE EXTREME DROITE	AI R7,-8	8 PIXELS A GAUCHE
NEG 2HB	REMPILSSAGE INVERSE POUR NOUVEAUX SECTEURS	RE155 BL 2REVS8R	LECTURE NOUVEL OCTET R7 (8 PIXELS A GAUCHE)
RE125 A 2GD,R10		CI R2,16	A GAUCHE DE L'ECRAN ?
C R10,R12		JLT RE160	160 POUR REMPLISSAGE A GAUCHE
JEQ RE130	BORNE DROITE ATTEINTE ?	MOV R9,R9	AUCUN PIXEL ALLUME DANS LES 8 ?
BL 2REBIT	PIXEL ALLUME ?	JNE RE160	160 POUR REMPLISSAGE A GAUCHE
JNE RE125	125:CONTINUER LA RECHERCHE	LI R10,2FF00	OCTET R7 VIDE:ALLUMER LES 8 PIXELS
MOV 2DAV,R12	BORNE GAUCHE POUR RECHERCHE EN SENS INVERSE	BL 2REVS8W	ALLUMER LES 8 PIXELS
DEC R10	(SUITE)SUR LE SEGMENT PRECEDENT	MOV8 2COLORS,R10	
MOV R10,2DAV	BORNE DROITE POUR SEGMENT SUIVANT	AI R7,2000	TABLE DES COULEURS
INC R10		BL 2REVS8W	COLORATION DES 8 PIXELS
A 2HB,R1	REPASSER SUR SEGMENT PRECEDENT:RECHERCHE CONTINUITE	AI R7,2E000	RETOUR A LA TABLE DES PATRONS
NEG 2GD	RECHERCHE VERS LA GAUCHE	AI R7,-8	8 PIXELS A GAUCHE
BL 2RESECT	NOUVEAUX SECTEURS ?	AI R2,-8	8 PIXELS A GAUCHE
NEG 2HB	REPRENDRE SENS REMPLISSAGE PRECEDENT	JMP RE155	POURSUITE REMPLISSAGE SEGMENT
A 2HB,R1	RETOURNER SUR SEGMENT PRECEDENT	RE160 LI R4,20001	PIXEL DROIT ALLUME DANS OCTET CORRESPOND A R7
JMP RE110	FIN DES CALCULS,REMPLIR LE SEGMENT SUIVANT	BL 2EXTREM	REMPILSSAGE EXTREME GAUCHE SEGMENT
RE135 MOV 2M1,260	DEBUT DE RECHERCHE BORNE VERS LA GAUCHE	INC R2	
LI R12,13	BORNE EXTREME GAUCHE	MOV R2,2GAP	NOUVELLE BORNE GAUCHE POUR SEGMENT SUIVANT
RE140 A 2GD,R10		MOV 2RTILIN,R11	
C R10,R12	BORNE GAUCHE ATTEINTE ?	B *R11	
JEQ RE145	145:EXTREME GAUCHE SEGME. SUR BORD GAUCHE ECRAN	*ALLUMAGE BIT DANS R4 CORRESPONDANT A POSITION R10	
BL 2REBIT	PIXEL ALLUME ?	REBIT MOV R11,2RTBIT	


```

MOV R10,R0      CALCUL DU NUMERO DE L'OCTET CORRESPONDANT A R10
MOV R1,R7
SLA R7,5
SOC R1,R7
ANDI R7,>FF07
MOV R0,R4
ANDI R4,7
A R0,R7
S R4,R7
BL @REVSBR
LI R4,>0080
ANDI R0,7
JEQ RE170
SRA R4,0
MOV @RTBIT,R11
S2C R9,R4
B *R11
**AFFICHAGE EXTREMITÉ DROITE A PARTIR DU BIT ALLUME DANS R4
EXTREM S2C R9,R4
JEQ RE180
SOC R4,R9
DEC R2
CI R2,12
JGT RE175
INC R2
JMP RE180
SLA R4,1
CI R4,>0100
JEQ RE180
JMP EXTREM
MOV R11,@RTBOUT *
MOV R9,R10
SWPB R10
BL @REVSBW
JMP RE185
REICOL MOV R11,@RTBOUT ENTREE POUR MODIF. COULEUR PIXEL DEJA ALLUME
RE185 AI R7,>2000 TABLE DES COULEURS
SWPB R7
MOV R7,@>8C02
SWPB R7
MOV R7,@>8C02
BL @REVSBW
JMP RE185
RE180 MOV R11,@RTBOUT *
MOV R9,R10
SWPB R10
BL @REVSBW
JMP RE185
RE175 SLA R4,1
CI R4,>0100
JEQ RE180
JMP EXTREM
MOV R11,@RTBOUT *
MOV R9,R10
SWPB R10
BL @REVSBW
JMP RE185
RE170 MOV R11,@RTBIT,R11
SRA R4,0
JEQ RE170
ANDI R0,7
MOV R4,>0080
BL @REVSBR
S R4,R7
A R0,R7
MOV R7,>FF07
ANDI R7,>FF07
SOC R1,R7
SLA R7,5
MOV R1,R7
MOV R10,R0

```

CR25	DEC R7	TEMPO PREMIERE TOUCHE
	JNE CR10	
	INC R9	
	JMP CR40	
CR30	DEC R8	TEMPO ATTENT NOUVELLE TOUCHE
	JNE CR10	
B	2CR140	SORTIE
CR35	CLR R9	REMISE A ZERO REPETITION
CR40	MOV8 2RTCLAV,R4	
	JNE CR45	X(LUTIN8)
	LI R3,7	
	JMP CR80	
CR45	CI R4,>0E00	C(LUTIN1)
	JNE CR50	
	CLR R3	
	JMP CR80	
CR50	CI R4,>0300	D(LUTIN2)
	JNE CR55	
	LI R3,1	
	JMP CR80	
CR55	CI R4,>0600	R(LUTIN3)
	JNE CR60	
	LI R3,2	
	JMP CR80	
CR60	CI R4,>0500	E(LUTIN4)
	JNE CR65	
	LI R3,3	
	JMP CR80	
CR65	CI R4,>0400	W(LUTIN5)
	JNE CR70	
	LI R3,4	
	JMP CR80	
CR70	CI R4,>0200	S(LUTIN6)
	JNE CR75	
	LI R3,5	
	JMP CR80	
CR75	CI R4,>0F00	Z(LUTIN7)
	JNE CR85	
	LI R3,6	
CR80	BL 2MODXY	
CR85	JMP CR05	

```
*TEST SUR LUTINS 1 A 8 RECHERCHE POSITION CRAYON (X/Y)
RTCR01 DATA 0 RETOUR A L'APPEL
RTCR02 DATA 0 RETOUR A L'APPEL
CRAY01 MOV R11,RTCR01 POUR RETOUR PROGRAMME PRINCIPAL
MOV 2SVCLAV,R4
```

JEQ CR90

MOV8 EYLUTIN+5,2ATRIBU+40

LI R0,TABLUT+40

LI R2,4

LI R8,2000 TEMPO NOUVELLE TOUCHE

BLWP 2KSCAN

JNE CR35 NOUVELLE TOUCHE TRAITEE DE SUITE (SANS REPETITION)

CI R4,>FF00 AUCUNE TOUCHE ?

CI R9,3 REPETITION ?

LI R7,200 VITESSE REPETITION

JNE CR20

```

*TRAITEMENT CRAYON
CR90 LI R2,4      4 BYTES A DEPLACER VERS VDP(TABLE ATTRIBUT LUTIN)
LI R10,63      TEMPORISATION
CLR R12      DARR=0 SENS TEST VERS LUTIN SUIVANT
LI R13,4
MOV8 2YLUTIN+1,2ATRIBU+32 ORDONNEE POUR MASQUE SUR DESSIN
CR95 LI R0,TABLUT ADRESSE VDP LUTIN 1
LI R1,ATRIBU ADRESSE LUTIN 1
CLR R3      POUR TEST SUR LUTIN 1
MOV R0,R7
MOV R1,R8
MOV R1,R8
CR100 LI R9,20      TEMPO. REPONSE CRAYON
CR105 LI R0,TABLUT+32 ADRESSE MASQUE DANS VDP
LI R1,ATRIBU+32
BLWP 2MBW      MASQUE SUR ECRAN
MOV R7,R0
MOV R8,R1
MOV8 2YLUTIN(R3),*R1 LUTIN (R3+1) SUR ECRAN
BLWP 2MBW
CLR R4
CR110 BLWP 2KSCAN
MOV8 2RTCLAV,R4
CI R4,>1200      CODE TIR (CRAYON ECLAIRE)
JEQ CR125      MODIFICATION XY
DEC R9
JNE CR110      BOUCLAGE POUR REPONSE CRAYON (KSCAN)
MOV R7,R0
MOV R8,R1
MOV8 2OUTECR,*R1 LUTIN TESTE MIS HORS ECRAN
BLWP 2MBW
DEC R10
JEQ CR130      FIN TEMPO. PLACEMENT CRAYON SUR ECRAN
MOV R12,R12      SENS DES TEST ?(LUTIN SUIVANT OU PRECEDENT)
JEQ CR120      TEST SUR LUTIN SUIVANT
CLR R12      TEST SUR LUTIN PRECEDENT PUIS SUIVANT
MOV R3,R3      TEST SUR LUTIN 1 ?
JEQ CR115      TEST SUR LUTIN 8(PRECEDENT)
DEC R3      TEST SUR LUTIN PRECEDENT
S R13,R7      LUTIN PRECEDENT
S R13,R8      ATTRIBUT LUTIN PRECEDENT
JMP CR100      REPRISE CYCLE DES TESTS

CR115 LI R0,TABLUT+28 ADRESSE LUTIN 8
LI R1,ATRIBU+28 ATTRIBUTS LUTIN 8
LI R3,7      TEST SUR LUTIN 8(R3+1)
MOV R0,R7
MOV R1,R8
JMP CR100      REPRISE CYCLE DES TESTS
CR120 CI R3,7      TEST 8 EN COURS ?
JEQ CR95      VERS TEST 1
INC R3      TEST LUTIN SUIVANT
A R13,R7      LUTIN SUIVANT
A R13,R8      ATTRIBUT LUTIN SUIVANT
JMP CR100      REPRISE CYCLE DES TESTS
CR125 BL 2MODXY
LI R10,39      TEMPORISATION
INC R12      DARR=1 POUR FUTUR TEST SUR LUTIN SUIVANT
LI R9,50      TEMPO
B 2CR105
CR130 LI R0,>8000      TEMPO POUR RETIRER LE CRAYON DE L'ECRAN
CR135 DEC R0
JNE CR135
MOV8 2OUTECR,2ATRIBU+32 ADRESSE MASQUE HORS ECRAN
LI R0,TABLUT+32
LI R1,ATRIBU+32
BLWP 2MBW      MASQUE HORS ECRAN
CR140 MOV 2RTCR1,R11
B *R11
*MODIFICATION POSITION X/Y CRAYON SUR ECRAN
MODXY MOV R11,2RTCR2
CLR R0
MOV8 2SENSXY(R3),R0 CODE DEPLACEMENT DU LUTIN NUMERO R3+1
MOV R0,R4
ANDI R4,>0200      DIRECTION VERS HAUT DE L'ECRAN ? (Y-1)
JNE CR145      SI NON TEST POUR Y+1
LI R6,YLUTIN ADRESSE ORDONNEE LUTIN 1
LI R1,-1      INCREMENT -1 POUR MODIF.Y
LI R4,>0300      LIMITE HAUTE DE L'ECRAN
LI R5,1      1 DANS R5 POUR MODIF.Y DES LUTINS (2Y+...)
BL 2CR175
CR145 MOV R0,R4
ANDI R4,>0100      DIRECTION VERS BAS DE L'ECRAN ? (Y+1)
JNE CR150

```

```

LI R6,YLUTIN
LI R1,1 INCREMENT +1 POUR MODIF.Y
LI R4,>8500 LIMITE BASSE DE L'ECRAN
LI R5,1 1 DANS R5 POUR MODIF Y DES LUTINS (2Y+...)
BL 2CR175
MOV R0,R4
ANDI R4,>2000 DIRECTION VERS LA GAUCHE DE L'ECRAN ? (X-1)
JNE CR155
LI R6,COORD+1 ADRESSE ABSCISSES POINT ET LUTINS
LI R1,-1 INCREMENT -1 POUR MODIFICATION X
LI R4,>0000 LIMITE GAUCHE DE L'ECRAN
LI R5,4 4 DANS R5 POUR MODIF.X DES LUTINS (2ATRIBU+...)
BL 2CR175
MOV R0,R4
ANDI R4,>1000 DIRECTION VERS LA DROITE DE L'ECRAN ? (X+1)
JNE CR160
LI R6,COORD+1
LI R1,1 INCREMENT +1 POUR MODIF.X
LI R4,>FC00 LIMITE DROITE DE L'ECRAN
LI R5,4 4 DANS R5 POUR MODIF. X DES LUTINS (2ATRIBU+...)
BL 2CR175
MOV R6,YLUTIN+8,2COORD NOUVELLE ORDONNEE Y DANS 2COORD
MOV R6,YLUTIN+1,2ATRIBU+32 NOUVELLE ORDONNEE POUR MASQUE
MOV R6,2ATRIBU+13,2ATRIBU+33 NOUVELLE ABSCISSE POUR MASQUE
MOV R0,2SVX
MOV R1,2SVY
MOV 2DP1,2DP1 DRAPEAU 1 TRACE CONTINU
JEQ CR165 DRAPEAU 0 TRACE POINT PAR POINT
BL 2POINT TRACE CONTINU
LWPI USRAS
JMP CR170
MOV 2DP4,2DP4 DRAPEAU REMPLISSAGE CURVILIGNE
JEQ CR170 REMPLISSAGE CURVILIGNE
INC 2DP2
BL 2LIGNE
CLR 2DP2
MOV 2RTCRA2,R11
B *R11
CR175 C8 *R6,R4 X OU Y SUR BORD DE L'ECRAN ?
JEQ CR185 SI OUI PAS DE MODIF. XY, RETOUR A L'APPEL
LI R4,9 SI NON, 9 LUTINS A DEPLACER (INCREMENT 1)

```

```

SWPB R1
CR180 AB R1,*R6 MODIF XY
A R5,R6 ADRESSE COORDONNEE SUIVANTE A MODIFIER
DEC R4 LUTIN SUIVANT
JNE CR180
CR185 B *R11

```

Source STOR/S

```

MODE DATA >0000 MODE LECTURE/ECRITURE
NOM DATA 304,328,280,320,328,296,400,0 "FICHIER"*(8 CARACTERES)
DATA 328,360,264,312,296,0,288,408,344 "IMAGE DSK"*(9 CARACTERES)
PAS DATA >80 NOMBRE DE BYTES PAR ENREGISTREMENT
PBUF DATA 0 ADRESSE BUFFER VDP
PDATA DATA >0000,>0000,>0001,PBUF,>8080,>0000,>000F
TEXT 'DSK.XXXXXXXXXX'
*TEST LECTURE/ECRITURE SUR DISQUE
STORE MOV 2DP8,2DP8 DRAPEAU ECRITURE/LECTURE
JEQ LECTU
LI R0,>0300 MODE ECRITURE
MOV R0,2MODE
JMP ST05
LECTU LI R0,>0200 MODE LECTURE
MOV R0,2MODE
*DEMANDE NOM FICHIER
ST05 CLR R3 NUMERO DE LA LETTRE A AFFICHER (DANS 2NOM)
LI R0,>1708 DEUXIEME CASE LIGNE 24
LI R2,8 8 BYTES PAR CARACTERE
LI R1,R1,TALPHA DEBUT TAMPONS CARACTERES ASCII
A 2NOM(R3),R1 ADRESSE DU CARACTERE A AFFICHER
BLWP 2NOM
AI R0,8 CASE SUIVANTE SUR LIGNE 24
INCT R3 LETTRE SUIVANTE
CI R3,34 FIN DU TAMPON 2NOM ?
JNE ST10
LI R0,0 CLAVIER ENTIER
MOV R0,2>8374
*SAISIE NOM DU FICHIER IMAGE
ST15 LI R2,8 8 BYTES A ECRIRE
LI R4,4 QUATRIEME CARACTERE DU NOM DU FICHIER (DSK.)

```



```

LI R0,>1790 CASE 19 LIGNE 24 (DISQUETTE 1 OU 2)
LI R1,TALPHA+504 SOULIGNE
BLWP 2MBW AFFICHAGE
ST20 CLR R3 VIDAGE R3 POUR STATUS (TOUCHE PRESSEE ?)
BLWP 2KSCAN APPEL CLAVIER
MOV8 2STATUS,R3 NOUVELLE TOUCHE PRESSEE ?
JEQ ST25 PAS DE TOUCHE PRESSEE
MOV8 2RTCLAV,R3 CODE TOUCHE PRESSEE (NUMERO DISK)
CI R4,4 SAISIE NUMERO DISK ?
JNE ST40 SAISIE NOM DE L'IMAGE
CI R3,>3100 CODE NOMBRE 1 ? (DSK1)
JNE ST30 POUR TEST SUR 2
LI R1,TALPHA+136 ADRESSE DU NOMBRE 1
BLWP 2MBW AFFICHAGE 1
JMP ST35 POUR TRANSFERT NUMERO DANS PDATA
ST30 CI R3,>3200 CODE NOMBRE 2 ? (DSK2)
JNE ST25 RETOUR CLAVIER REFUSE (1 OU 2 VALABLE)
LI R1,TALPHA+144 ADRESSE DU NOMBRE 2
BLWP 2MBW AFFICHAGE 2
ST35 MOV8 R3,2PDATA+17 NUMERO LECTEUR DISQUETTE DANS PDATA
LI R1,TALPHA+112 POINT
AI R0,8 CASE 20 LIGNE 24
BLWP 2MBW AFFICHAGE DU POINT
LI R4,6 SIXIEME LETTRE DU NOM DU FICHIER (DSKX.?)
AI R0,8 CASE 21 LIGNE 24
JMP ST20 APPEL PREMIER CARACTERE DU NOM DE L'IMAGE
ST40 CI R3,>0800 TOUCHE RETOUR A GAUCHE ?
JNE ST45 TEST SUR CARACTERE
CI R4,6 PREMIERE LETTRE DU NOM DU FICHIER ?
JEQ ST15 REPRISE DES SAISIES AU DEBUT
CI R4,4 PREMIER CARACTERE ?
JEQ ST25 SOULIGNE DEJA A GAUCHE
LI R1,TALPHA CARACTERE ESPACE (EFFACEMENT SOULIGNE)
BLWP 2MBW EFFACEMENT SOULIGNE
DEC R4 CARACTERE PRECEDENT
AI R0,-8 CASE PRECEDENTE A L'ECRAN
JMP ST20 APPEL NOUVEAU CARACTERE DE REMPLACEMENT
ST45 CI R4,6 TEST PREMIER CARACTERE (LETTRE ?)
JNE ST50 TEST VALIDITE CARACTERE (POINT/ESPACE REFUSE)
CI R3,>4100 LETTRE A
JLT ST25 PREMIER CARACTERE REFUSE

```

```

CI R3,>5A00 LETTRE Z
JGT ST25 PREMIER CARACTERE REFUSE
ST50 CI R3,>2000 CARACTERE ESPACE INTERDIT
JEQ ST25 REFUSE (APPEL NOUVELLE TOUCHE)
CI R3,>2E00 CARACTERE POINT INTERDIT
JEQ ST25 REFUSE (APPEL NOUVELLE TOUCHE)
CI R3,>0D00 TOUCHE ENTER (FIN DU NOM)
JEQ ST55 FIN DE LA SAISIE DU NOM DU FICHIER
MOV8 R3,2PDATA+13(R4) CARACTERE DANS 2PDATA
LI R1,TALPHA ADRESSE TAMPON CARACTERES ASCII
SWPB R3 CODE ASCII DANS BYTE FAIBLE
AI R3,-32 CODE ASCII-32 (NUMERO D'ORDRE DANS TAMPON T)
SLA R3,3 8*R3 POUR ADRESSE DU CARACTERE DANS TAMPON T
A R3,R1 ADRESSE DU CARACTERE DANS TAMPON T
BLWP 2MBW AFFICHAGE DU CARACTERE
CI R4,15 DIXIEME CARACTERE ?
JEQ ST25 CORRECTION DERNIER CARACTERE ?
INC R4 CARACTERE SUIVANT
AI R0,8 CASE SUIVANTE A L'ECRAN (LIGNE 24)
JMP ST20 APPEL PROCHAIN CARACTERE
ST55 DEC R4 TRANSFERT LONGUEUR DU NOM FICHIER DANS PDATA
SWPB R4 VALEUR DANS BYTE FORT
MOV8 R4,2PDATA+13 LONGUEUR DU NOM DU FICHIER DANS 2PDATA
LI R0,>0100 DEMI-CLAVIER GAUCHE
MOV8 R0,>8374
*ENREGISTREMENT/LECTURE IMAGE SUR DISQUE
*OUVERTURE FICHIER
LI R0,PAB PAB MODE OUVERTURE
LI R1,PDATA
LI R2,>20
BLWP 2MBW PAB DANS LA VDP
LI R6,PAB+13
MOV R6,2PNTR
BLWP 2DSRLNK OUVERTURE DU FICHIER
DATA 8
*ECRITURE/LECTURE SUR DISQUE
MOV8 2MODE,R1 PAB AU MODE ECRITURE/LECTURE
LI R0,PAB+4
BLWP 2VSBW
LI R0,PAB+6
CLR 2PBUF ADRESSE DE LA TABLE DES PATRONS

```

```

LI R1,PBUF
LI R2,2
BLWP 2MBW
CLR R3          NUMERO D'ENREGISTREMENT
ST60 MOV R6,2PNTR
      BLWP 2DSRLNK
      DATA 8
      INC R3
      A 2PAS,2PBUF
      BLWP 2MBW
      CI R3,46
      JNE ST60
LI R1,2000 ADRESSE DE LA TABLE DES COULEURS
MOV R1,2PBUF
LI R1,PBUF
BLWP 2MBW
ST65 MOV R6,2PNTR
      BLWP 2DSRLNK
      DATA 8
      INC R3
      A 2PAS,2PBUF
      BLWP 2MBW
      CI R3,92
      JNE ST65
*FERMETURE FICHIER
LI R1,20100 PAB AU MODE FERMETURE
LI R0,PAB+4
BLWP 2USBW
MOV R6,2PNTR
BLWP 2DSRLNK
DATA 8          FERMETURE DU FICHIER
*EFFACEMENT LIGNE 24
LI R3,31
LI R0,21708 +PREMIERE CASE LIGNE 24
LI R1,TALPHA CARACTERE ESPACE
LI R2,8
BLWP 2MBW
AI R0,8          CASE SUIVANTE
DEC R3
JNE ST70         DERNIERE CASE ?
B *R11
ST70

```

Source MENU/S

```

$MENU DATA 0 SAUVEGARDE MENU EN COURS
RTTM DATA 0 RETOUR A L'APPEL
MENU10 BYTE 32,32,32,32,32,32
      BYTE 32,84,82,65,73,50      TRA12
      BYTE 32,82,69,67,84,51      RECT3
      BYTE 32,67,85,82,86,52      CURV4
      BYTE 32,85,84,73,76,53,32,70 UTILS F(FIN D'EXECUTION)
      DATA MENU10,MENU10,MENU10,MENU10,MENU10,MENU10,MENU10,MENU10
      DATA RESE01,SAUT2,SAUT2,SAUT2,SAUT2,SAUT2,SAUT2,SAUT2
MENU100 BYTE 32,32,32,32,32,32
      BYTE 32,32,32,32,32,32
      BYTE 32,32,32,32,32,32
      BYTE 32,32,32,32,32,32
      BYTE 32,32,32,32,32,32
      BYTE 32,32,32,32,32,32,84      T(MENU)
      DATA MENU100,MENU100,MENU100,MENU100,MENU100,MENU100,MENU100,MENU100
      DATA RESE01,RESE01,RESE01,RESE01,RESE01,RESE01,RESE01,SAUT2
MENU110 BYTE 32,86,73,83,73,49      VISI1
      BYTE 32,69,70,70,65,50      EFFA2
      BYTE 32,67,79,85,76,51      COUL3
      BYTE 32,70,79,78,68,52      FOND4
      BYTE 32,84,89,80,69,53,32,84 TYPES T
      DATA MENU110,MENU110,MENU10,MENU10,MENU10,MENU10,MENU10,MENU10
      DATA RVISI8,REFFAC,RCOULE,RFOND,RESE01,RESE01,SAUT2
MENU120 BYTE 32,65,88,69,83,49      AXES1
      BYTE 32,83,69,71,77,50      SEG2
      BYTE 32,68,82,79,73,51      DR013
      BYTE 32,82,69,76,73,52      REL14
      BYTE 32,80,79,73,78,53,32,84 POIN5 T
      DATA MENU120,MENU120,MENU10,MENU10,MENU10,MENU10,MENU10,MENU10
      DATA RAXES,RLIGNE,SAUT2,RESE01,RPOINT,SAUT2
MENU121 BYTE 32,72,79,82,73,49      HORI1
      BYTE 32,86,69,82,84,50      VERT2
      BYTE 32,79,82,73,69,51      ORIE3
      BYTE 32,68,69,77,73,52      DEM14
      BYTE 32,32,32,32,32,32,84      T
      DATA MENU121,MENU10,MENU10,MENU10,MENU10,MENU10,MENU10,MENU10
      DATA RHORI2,RVERT1,RESE01,RESE01,RESE01,RESE01,SAUT2
MENU130 BYTE 32,34,79,78,34,49      "ON"1
      BYTE 32,34,79,70,34,50      "OF"2

```



```

JMP ME70
ME50 C1 R5,>0A00 TOUCHE 5
JNE ME55
LI R6,4
JMP ME70
ME55 C1 R5,>0B00 TOUCHE T
JNE ME60
LI R6,5
JMP ME70
ME60 DEC R4
JNE ME30
INC R6
C1 R6,5 OPTION SUIVANTE
JLT ME20 AVANT/DERNIERE OPTION ?
JNE ME65 OPTION SUIVANTE
BL 2MASQ PLUS D'OPTION
B 2ME80 EFFACEMENT LIGNE 24
LI R7,>F800 PLUS D'OPTION A TESTER
MOV R7,2ATRIBU+49 *
MOV R7,2ATRIBU+53 * 5 LUTINS SUR OPTION MENU (T)
MOV R7,2ATRIBU+57 *
MOV R7,2ATRIBU+61 *
BLWP 2MBW
JMP ME25 DERNIER TEST OPTION M (MENU)
BL 2MASQ EFFACEMENT MASQUE 4 CASES
MOV 2SMENU,R4 ADRESSE MENU EN COURS
AI R4,32 PREMIERE LIGNE DATA DANS TABLEUX MENU
SLA R6,1 2*R6 (NUMERO D'ORDRE DU PROCHAIN MENU)
A R6,R4 ADRESSE DU PROCHAIN MENU DANS TABLEUX
MOV *R4,2SMENU SAUVEGARDE DU PROCHAIN MENU
AI R4,12 DEUXIEME LIGNE DATA DANS TABLEUX MENU
MOV *R4,R1 ADRESSE DE LA ROUTINE A EXECUTER
B *R1 BRANCHEMENT POUR EXECUTION ROUTINE (DU MENU)
RPOINT BL 2POINT ROUTINE POINT
LWPI USRWIS
JMP SAUT2
RAXES BL 2HORIZ ROUTINE HORIZONTALE
BL 2VERTI ROUTINE VERTICALE
JMP SAUT2
RLIGNE BL 2LIGNE ROUTINE SEGMENT
JMP SAUT2

```

```

RSTORE C1 R6,2 LECTURE ? (OPTION MENU 2:R6=2)
JEQ RSTOR1
INC 2DP8 DRAPEAU ECRITURE DANS FICHIER
JMP RSTOR2
RSTOR1 CLR 2DP8 DRAPEAU LECTURE DANS FICHIER
RSTOR2 BL 2STORE ROUTINE LECTURE/ENREGISTREMENT SUR DISQUE
JMP SAUT2
RCOULE BL 2NEWCOL CHOIX NOUVELLE COULEUR
JMP SAUT2
RFOND MOV 2DF6,2DP6
JEQ RFOND1
CLR 2DP6 ANNULATION REMPLISSAGE FOND
CLR 2DP1 TRACE CRAYON NON VISIBLE
LI R12,MEN110
MOV R12,2MEN110+38
MOV 2SVCOL,2COLORS
JMP SAUT2
RFOND1 INC 2DP6 POUR REMPLISSAGE FOND
INC 2DP1 TRACE CONTINU
MOV 2COLORS,2SVCOL
BL 2NEWCOL CHOIX COULEUR DE FOND
LI R12,MEN110
MOV R12,2MEN110+38
JMP SAUT2
RVISIB MOV 2DP1,2DP1 DRAPEAU TRACE VISIBLE ?
JNE RINVIS
INC 2DP1 DEPLACEMENT VISIBLE DU CRAYON
MOV 2DP3,2DP3
JNE VISIB2
VISIB1 LI R12,MEN110
MOV R12,2MEN110+32
MOV R12,2MEN110+34
JMP SAUT2
RINVIS CLR 2DP1 SUPPRESSION DEPLACEMENT VISIBLE DU CRAYON
MOV 2DP3,2DP3
JNE VISIB1
JMP VISIB4
REFFAC MOV 2DP3,2DP3 CHOIX D'EFFACEMENT AVEC CRAYON
JNE VISIB3
INC 2DP3 DRAPEAU=1 : SUPPRESSION DE L'EFFACEMENT
MOV 2DP1,2DP1

```



```

JEQ VIS181
VIS182 LI R12,MENUI0
MOV R12,MEN110+32
LI R12,MEN110
MOV R12,MEN110+34
SAUT2 JMP ME75
VIS183 CLR ZDP3
MOV ZDP1,ZDP1
JNE VIS181
VIS184 LI R12,MEN110
MOV R12,MEN110+32
LI R12,MENUI0
MOV R12,MEN110+34
JMP ME75
RECRAN BL ZEFFT
RESE01 JMP ME75
RON CLR ZDP6
JMP ME75
ROFF INC ZDP6
JMP ME75
RCSURF MOV ZDP5,ZDP5
JEQ RCSUR1
LI R12,MEN130
MOV R12,MEN130+38
CLR ZDP6
CLR ZDP5
CLR ZDP4
MOV ZSVCOL,ZCOLORS
JMP ME75
RCSUR1 LI R12,MENUI0
MOV R12,MEN130+38
INC ZDP5
MOV ZCOLORS,ZSVCOL
JMP ME75
RCSUR2 MOV ZDP5,ZDP5
JEQ ME75
INC ZDP4
INC ZDP3
CLR ZDP1
BL ZNEWCOL
BL ZLIGNE

```

DRAPEAU=0 : EFFACEMENT AU CRAYON

EFFACEMENT DE TOUT L'ECRAN

```

JMP ME75
RHORIZ MOVB ZCOORD+1,R1
MOVB ZP1,ZCOORD+1
BL ZHORIZ
MOVB R1,ZCOORD+1
JMP ME75
RVERTI MOVB ZCOORD,R1
MOVB ZOUTECL,ZCOORD
BL ZVERTI
MOVB R1,ZCOORD
JMP ME75
RSURFA MOV ZCOLORS,ZSVCOL
BL ZNEWCOL
BL ZREPLI
MOV ZSVCOL,ZCOLORS
ME75 NEG ZSVCLAV INVERSION RTCLAV POUR TEST DANS PROG
ME80 MOV ZSMENU,ZMENU MENU EN COURS
BL ZAFFMEN AFFICHAGE MENU EN COURS
MOV ZRTTM,R11
LI R1,>8000 TEMPORISATION
ME85 DEC R1
JNE ME85
B *R11
* AFFICHAGE MENU (ADRESSE MENU DANS SAVMEN)
AFFMEN MOV ZMENU,R5 ADRESSE DU MENU
LI R4,32 31 CARACTERES A AFFICHER
LI R0,>1700 CASE 2 POUR AFFICHAGE
LI R2,8 8 BYTES A AFFICHER
ME90 LI R1,TALPHA ADRESSE DES CARACTERES ASCII
CLR R3
MOVB *R5,R3 CODE ASCII DE LA LETTRE
SWPB R3 CODE ASCII DANS BYTE FAIBLE
AI R3,-32 NUMERO LETTRE DANS TAMPON T
SLA R3,3 *=8 (ADRESSE LETTRE DANS T)
A R3,R1 ADRESSE LETTRE DANS R1
BLWP ZMBW AFFICHAGE DE LA LETTRE
INC R5 BYTE SUIVANT (LETTRE SUIVANTE)
AI R0,8 ADRESSE CSAE SUIVANTE SUR LIGNE 24
DEC R4 LETTRE SUIVANTE
JNE ME90 SI NON LETTRE SUIVANTE
LI R2,16

```

```

B *R11
* EFFACEMENT MASQUE 4 CASES
MASQ LI R0,TABLUT+48 LUTIN CARRE
LI R1,ATRIBU+48 POSITION PREMIER LUTIN
LI R2,16 4 LUTINS (16 BYTES D'ATTRIBUT)
CLR R4
MOV8 2OUTEGR,R4 POSITION Y HORS ECRAN
MOV8 R4,2ATRIBU+48 LUTIN 1
MOV8 R4,2ATRIBU+52 LUTIN 2
MOV8 R4,2ATRIBU+56 LUTIN 3
MOV8 R4,2ATRIBU+60 LUTIN 4
BLWP 2MBW
B *R11

```

EFFACEMENT DES 4 LUTINS

Source NEWC/S

```

** NEWC/S **
RTNC DATA 0 POUR RETOUR APPEL
NEWCOL MOV R11,2RTNC
MOV 2MASQUE,2MENU POUR LIGNE 24 EN NOIR
BL 2AFFMEN

* MISE EN PLACE DES COULEURS DANS TABLE/COULEURS
NE05 CLR R3 NUMERO DU CARRE AFFICHE
LI R0,>3708 ADRESSE DEUXIEME CASE
NE10 MOV 2TABCOL(R3),R1 ADRESSE NOUVELLE COULEUR
CLR R4 NUMERO DU BYTE (6 BYTES COULEUR, 2 BLANC)
NE15 BLWP 2VSBW
INC R0 BYTE SUIVANT
INC R4
C1 R4,6 DERNIER BYTE COULEUR ?
JNE NE15
AI R0,8 SAUT D'UNE CASE
INCT R0 DEBUT DE LA CASE
INCT R3
C1 R3,28 DERNIERE COULEUR ?
JNE NE10

* AFFICHAGE DES 14 CARRES COULEURS
LI R0,>1708 ADRESSE CASE 2
LI R2,8 8 BYTES A CHARGER

```

```

NE20 LI R1,LUTINS+88 CARACTERE PLEIN
BLWP 2MBW
AI R0,16 SAUT D'UNE CASE
DECT R3 NUMERO CASE SUIVANTE
JNE NE20 DERNIERE CASE ?

* AFFICHAGE N(NOIR) ET T(TRANSPARENT)
LI R0,>17E8 CASE 30
LI R1,TALPHA+416 LETTRE T
LI R2,8
BLWP 2MBW
LI R0,>17F8 CASE 32
LI R1,TALPHA+368 LETTRE N
BLWP 2MBW

* AFFICHAGE DES 14 REPERES COULEURS
LI R0,>1710 ADRESSE CASE 3
LI R2,8 8 BYTES A CHARGER
LI R3,28
LI R1,TALPHA+256 LETTRE 2
NE25 BLWP 2MBW
AI R0,16 SAUT D'UNE CASE
AI R1,8
DECT R3 NUMERO CASE SUIVANTE
JNE NE25 DERNIERE CASE ?

* ATTENTE CRAYON ECLAIRE
LI R4,>8000 TEMPO POUR PLACER LE CRAYON
NE30 DEC R4
JNE NE30
CLR R3
MOV8 R3,2>8374 SELECTION DE TOUT LE CLAVIER
BLWP 2KSCAN
MOV8 2RTCLAV,R3
C1 R3,>4000 LETTRE 2
JLT NE55
C1 R3,>4E00 LETTRE N
JLT NE45
JGT NE40
LI R3,15 COULEUR NOIRE
JMP NE50
C1 R3,>5400 LETTRE T
JNE NE55
LI R3,16 COULEUR TRANSPARENT

```

```

NE45 JMP NE50
LI R4,>4000
S R4,R3
SWPB R3
NE50 LI R4,>0100
MOV8 R4,>8374
JMP NE85
NE55 LI R4,>0100 SELECTION CLAVIER GAUCHE
MOV8 R4,>8374
CLR R4
BLWP 2KSCAN TEST SUR CRAYON
MOV8 2RTCLAV,R4 RETOUR CRAYON
CI R4,>1200 CRAYON ECLAIRE ?
JNE NE35 PAS DE RETOUR CRAYON
* CONTROLE CRAYON SUR LIGNE 24
MOV 2MASQUE,2MENU
BL 2AFFMEN
LI R4,>2000 TEMPO POUR EXTINCTION CRAYON
DEC R4
JNE NE60 FIN DE TEMPO ?
LI R3,25 TEMPO POUR REPONSE CRAYON
BLWP 2KSCAN APPEL CRAYON
MOV8 2RTCLAV,R4 RETOUR CRAYON
CI R4,>1200 S1 CRAYON ENCORE ECLAIRE,MAUVAIS RETOUR CRAYON
JNE NE70 RECOMMENCER LA ROUTINE
B 2NE05
NE70 DEC R3
JNE NE65 FIN DE TEMPO ?
* SEQUENCE RECHERCHE COULEUR CHOISIE
MOV8 2MENU,2ATRIBU+60 ORDONNEE LUTIN 4 (SUR LIGNE 24)
LI R0,>0800 CASE 2 LIGNE 24
MOV8 R0,2ATRIBU+61 ABSCSSE LUTIN 4 (SUR LIGNE 24)
LI R0,TABLUT+60 PATRON LUTIN 4 (BLANC SUR CASE TESTEE)
LI R1,ATRIBU+60 ATTRIBUTS LUTIN 4
BLWP 2MBW AFFICHAGE CARRE
LI R5,25 TEMPO REPONSE CRAYON
BLWP 2KSCAN APPEL CRAYON
MOV8 2RTCLAV,R4 RETOUR CRAYON
CI R4,>1200 CRAYON ECLAIRE ?
JEQ NE85 COULEUR RETENUE
DEC R5 FIN DE TEMPO ?

```

```

JNE NE80 NOUVEAU TEST CRAYON
LI R5,>1000 16 BYTES SUPPLEMENTAIRES (2 CASES A DROITE)
AB R5,2ATRIBU+61 2 CASES A DROITE
INC R3 NUMERO DE CASE SUIVANT
CI R3,16 DERNIERE CASE ?
JNE NE75 TEST NOUVELLE CASE
MOV8 2OUTEGR,2ATRIBU+60 LUTIN 4 HORS ECRAN
BLWP 2MBW EFFACEMENT LUTIN 4
B 2NE05 PAS TROUVE DE CASE (RETOUR DEBUT ROUTINE)
NE85 MOV8 2OUTEGR,2ATRIBU+60 LUTIN 4 HORS ECRAN
BLWP 2MBW EFFACEMENT LUTIN 4
NE90 INC R4 REPONSE ACCEPTEE (TEMPO)
CI R4,>5000
JNE NE90 FIN DE TEMPO ?
SLA R3,1 2*R3
MOV 2TABCOL(R3),2COLORS NOUVELLE COULEUR
* COULEUR CARACTERES ASCII
LI R0,>3700
LI R1,>F100
NE95 BLWP 2VSBW
INC R0
CI R0,>3800
JNE NE95
MOV 2RTNC,R11 RETOUR A L'APPEL
B *R11

```

Source GRAPH/S

```

RTGRAF DATA 0
CR5 DATA 0,0,0,0
*ROUTINE TRACE DE POINT
POINT MOV R11,2RTGRAF APPEL AVEC Y/X DANS COORD
LWPI GRAFWS
MOV 2RTGRAF,R11
MOV8 2COORD+1,R0 X DANS R0
SRL R0,8
MOV8 2COORD,R1 Y DANS R1
SRL R1,8
JMP POINT3

```

*APPEL DEPUIS ROUTINE GRAPHIQUE CALCULANT UNE COURBE,Y/X DANS SUY/SUX

```

POINTC MOV R11,2RTGRAF
LWPI GRAFWS
MOV 2RTGRAF,R11
MOV 2SUX,R0
MOV 2SVY,R1
CI R0,13
JLT GR20
CI R0,252
JGT GR20
CI R1,3
JLT GR20
CI R1,181
JGT GR20
JMP POINT3
POINT2 MOV R11,2RTGRAF APPEL AVEC Y/X DANS SVY/SUX
LWPI GRAFWS
MOV 2RTGRAF,R11
MOV 2SUX,R0
MOV 2SVY,R1
POINT3 MOV R1,R4
SLA R4,5
SOC R1,R4
ANDI R4,>FF07
MOV R0,R5
ANDI R5,7
A R0,R4
S R5,R4
MOV 2DP6,2DP6
JEQ GR05
LI R5,>4000
AI R4,>2000
SWPB R4
MOV R4,>8C02
SWPB R4
MOV R4,>8C02
MOV 2COLORS,R0
MOV 2>8800,R6
ANDI R6,>F000
ANDI R0,>F000
SRL R0,4
SOCB R0,R6
JMP GR15
NUMERO DE L'OCLET DANS R4
REMPLISSAGE FOND ?
GR05:PAS DE REMPLISSAGE DU FOND
=VSBR
MOV R1,>8C00
SWPB R4
MOV R4,>8C02
SWPB R4
MOV 2COLORS,R0
MOV 2>8800,R6
ANDI R6,>F000
ANDI R0,>F000
SRL R0,4
SOCB R0,R6
JMP GR15
EN REMPLACEMENT DE "NOP"
GARDER COULEUR "ON"
COULEUR "ON"
TRANSFORMEE EN COULEUR "OFF"
COULEURS "ON"+"OFF"

```


*ROUTINE TRACE D'AXES

HORIZ LWPI GRAFWS

MOVB 2COORD,R9 Y

SRL R9,8

LI R8,13

DEBUT A GAUCHE DE L'ECRAN

MOV R8,R0

MOV R9,R1

BL 2POINT3

INC R8

C1 R8,253

COTE DROIT DE L'ECRAN ?

JLT GR25

JMP GR35

VERTI LWPI GRAFWS

MOVB 2COORD+1,R8 X

SRL R8,8

LI R9,3

DEBUT EN HAUT DE L'ECRAN

MOV R8,R0

MOV R9,R1

BL 2POINT3

INC R9

C1 R9,182

BAS DE L'ECRAN ?

JNE GR30

LWPI USRWS

B *R11

*ROUTINE TRACE DE SEGMENT

LIGNE MOV 2DP2,2DP2

JNE GR40

MOV 2COORD,2LIGNES

MOV 2P1,2DP2

1:DEUXIEME POINT DU SEGMENT

0:PREMIER POINT DU SEGMENT

1 POUR ATTENTE DEUXIEME POINT

MOV 2YLUTIN+5,2ATRIBU+36 Y POUR AFFICHAGE MIRE FIXE SEGMENT

MOV 2ATRIBU+13,2ATRIBU+37 X POUR AFFICHAGE MIRE FIXE SEGMENT

LI R0,TABLUT+36

LI R1,ATRIBU+36

LI R2,4

AFFICHAGE MIRE FIXE

BLWP 2MBW

B *R11

GR40 LWPI GRAFWS

MOV 2OUTECR,2ATRIBU+36

LI R0,TABLUT+36

LI R1,ATRIBU+36

LI R2,4

```

BLWP 2MBW
MOV 2COORD,2LIGNES+2
CLR 2DP2
MOV 2LIGNES+1,R9
SRL R9,8
MOV 2LIGNES+3,R7
SRL R7,8
MOV R7,2CR5+4
MOV 2LIGNES,R10
SRL R10,8
INC R10
MOV 2LIGNES+2,R8
SRL R8,8
INC R8
MOV R8,2CR5+6
CLR R12
MOV 2P1,2CR5
MOV 2P1,2CR5+2
S R9,R7
JLT GR45
JMP GR50
MOV 2M1,2CR5
NEG R7
MOV R7,R7
JNE GR55
SET0 R12
S R10,R8
JLT GR60
JMP GR65
MOV 2M1,2CR5+2
NEG R8
MOV R9,R0
MOV R10,R1
DEC R1
BL 2POINT3
C R9,2CR5+4
JNE GR70
C R10,2CR5+6
JEQ GR80
MOV R12,R12
JLT GR75

```

```

A 2CR5,R9
S R8,R12
JMP GR65
GR75 A 2CR5+2,R10
A R7,R12
JMP GR65
GR80 LWP1 USRWIS
B *R11
EFTT L1 R0,>2000 ADRESSE TABLE DES COULEURS
L1 R1,>1100 NOIR SUR NOIR (LES POINTS RESTENT ALLUMES)
GR85 BLWP 2VSBW
INC R0
C1 R0,>3700 23 LIGNES A COLORIER NOIR SUR NOIR
JNE GR85
CLR R0 DEBUT DE LA TABLE DES PATRONS
CLR R1 BYTE FORT=0
GR90 BLWP 2VSBW
INC R0
C1 R0,>1700 23 LIGNES A ETEINDRE
JNE GR90
B *R11

```

Source PR06/5

```

** PROG/5 ** SUIVI CRAYON SUR ECRAN **
CRAYON LWP1 USRWIS ESPACE DE TRAVAIL PROGRAMME PRINCIPAL
BL 2MOD81T PASSAGE EN MODE BIT/MAP
BL 2CHARLU CHARGEMENT DES LUTINS DANS LES TABLES VDP
BL 2TESMEN AFFICHAGE PREMIER MENU
PR05 CLR 2DP0 DRAPEAU 0:FLASH ETEINT
BL 2FLASH FLASH ETEINT
BL 2MIRE MIRE AFFICHEE
PR10 LIMI 2 INTERRUPTION ?
LIMI 0
INC 2DP0 DRAPEAU 1:FLASH ALLUME
BL 2FLASH MIRE ETEINTE,FLASH ALLUME
L1 R1,100 TEMPORISATION POUR REACTION CRAYON SUR FLASH
PR15 DEC R1
JEQ PR05 POUR NOUVEL APPEL CRAYON OU CLAVIER
BLWP 2KSCAN APPEL CRAYON OU CLAVIER

```

```

CLR R0
MOV 2RTCLAV,R0
C1 R0,>FF00 TOUCHE ENFONCEE ?
JEQ PR15
MOV R0,2SVCLAV SAUVEGARDE R0
BL 2TESMEN TEST CRAYON SUR LIGNE 24 ?
MOV 2SVCLAV,2SVCLAV VALEUR CHANGEE PAR TESMEN ?
JLT PR10 SVCLAV<0->UNE OPTION MENU A ETE EXECUTEE
CLR 2DP0 DRAPEAU POUR ETEINDRE LE FLASH ET POUR RECHERCHER
BL 2FLASH LA DIRECTION DE DEPLACEMENT DU CRAYON,
BL 2MIRE FLASH ET MIRE ETEINTS
BL 2CRAY01 RECHERCHE DIRECTION CRAYON
JMP PR05 RETOUR NOUVELLE SEQUENCE MIRE/FLASH
MIRE MOV 2YLUTIN+5,2ATRIBU+40 ORDONNEE POUR MIRE
MOV 2ATRIBU+13,2ATRIBU+41 ABSCISSE POUR MIRE
L1 R0,TABLUT+40 ADRESSE MIRE DANS TABLE D'ATTRIBUTS
L1 R1,ATRIBU+40 NOUVEAUX ATTRIBUTS MIRE
L1 R2,>0004 4 BYTES A CHARGER
BLWP 2MIBW NOUVELLE POSITION MIRE DANS TABLE D'ATTRIBUTS
L1 R3,20000 TEMPORISATION MIRE A L'ECRAN
PR20 DEC R3
NOP
JNE PR20
MOV 2OUTECR,2ATRIBU+40 FIN DE TEMPO. -> MIRE HORS ECRAN
BLWP 2MIBW
B *R11
FLASH MOV 2DP0,2DP0 DRAPEAU 0 -> FLASH A ETEINDRE
JEQ PR25
MOV 2YLUTIN+5,2ATRIBU+44 SI NON,ALLUMER FLASH:Y POUR FLASH
MOV 2ATRIBU+13,2ATRIBU+45 A L'ECRAN
JMP PR30
PR25 MOV 2OUTECR,2ATRIBU+44 FLASH HORS ECRAN
PR30 L1 R0,TABLUT+44 ADRESSE FLASH DANS TABLE D'ATTRIBUTS
L1 R1,ATRIBU+44 NOUVEAUX ATTRIBUTS FLASH
L1 R2,>0004 4 BYTES A CHARGER
BLWP 2MIBW
B *R11
* TABLE D'ECRAN
MODBIT L1 R0,>01F0 >F0 DANS R1 VDP POUR INHIBER L'ECRAN PENDANT LES
BLWP 2WATR OPERATIONS DE MISE EN PLACE DU MODE BIT-MAP
L1 R0,>1C00 ADRESSE TABLE D'ECRAN

```

```

CLR R2
PR35 CLR R1      DEBUT D'UN DES TROIS TIERS DE LA TABLE D'ECRAN
CLR R3
PR40 BLWP 2WSBW  NUMERO DE L'OCTET D'ECRAN DANS LA TABLE
INC R0
INC R3
AI R1,>0100      NUMERO SUIVANT POUR OCTET SUIVANT DANS LA TABLE
CI R3,>0100      FIN DU TIERS DE L'ECRAN ?
JNE PR40
INC R2
CI R2,>0003      TIERS SUIVANT
JNE PR35          TROISIEME TIERS TERMINE ?

* TABLE DES PATRONS
CLR R0
CLR R1
PR45 BLWP 2WSBW  ADRESSE DE LA TABLE DES PATRONS
INC R0            TOUS LES BYTES SONT ETEINTS
CI R0,>1800      MISE A 0 DE TOUTE LA TABLE DES PATRONS (24 LIGNES)
JNE PR45

* TABLE DES COULEURS
LI R0,>2000      DEBUT DE LA TABLE DES COULEURS
LI R1,>1100      NOIR SUR NOIR
BLWP 2WSBW
INC R0
CI R0,>3700      23 LIGNES COLORIEES (LIGNE 24 POUR MENU/QUESTIONS)
JNE PR50

* COULEUR DES CARACTERES ASCII (LIGNE 24 A L'ECRAN)
LI R0,>3700      ADRESSE AFFICHAGE CARACTERES ASCII (LIGNE 24)
LI R1,>F100      CARACTERES ASCII NOIR SUR BLANC
BLWP 2WSBW
INC R0
CI R0,>3800      FIN DE LA LIGNE MENU (LIGNE 24)
JNE PR55

*MODIFICATION DES REGISTRES VDP
LI R0,>0002      R0=02 (-) BIT6=1 :MODE M3 (BIT-MAP MODE)
BLWP 2WATR
LI R0,>01E0      R1=E0=80+40+20 (-) BITS 0,1,2=1
BLWP 2WATR      16KRAM,DISPLAY SCREEN,VDP INTERRUPT
LI R0,>0207      R2=07 (-) 07*400=1C00
BLWP 2WATR      ADRESSE TABLE D'ECRAN:>1C00
LI R0,>03FF      R3=FF (-) FF*40=...>2000 !!!

```

```

BLWP 2WATR      ADRESSE TABLE DES COULEURS:>2000
LI R0,>0403      R4=03 (-) 03*800=...>0000 !!!
BLWP 2WATR      ADRESSE TABLE DES PATRONS:>0000
LI R0,>0537      R5=37 (-) 37*80=1B80
BLWP 2WATR      ADRESSE TABLE D'ATTRIBUTS DES LUTINS:>1B80
LI R0,>0603      R6=03 (-) 03*800=1800
BLWP 2WATR      ADRESSE TABLE DE DESCRIPTION DES LUTINS:>1800
LI R0,>0701      R7=01 (-) ECRAN NOIR
BLWP 2WATR
B #R11

*LUTINS
CHARLU LI R3,>0100 SELECTION DEMI-CLAVIER GAUCHE
MOV8 R3,2>8374
LI R0,>1800      TABLE DE DESCRIPTION DES LUTINS
LI R1,LUTINS     ADRESSE DES DONNEES DES FORMES DES LUTINS
LI R2,152        152 BYTES A CHARGER (19 PATRONS)
BLWP 2WMBW
LI R0,>1B80      TABLE DES ATTRIBUTS DES LUTINS
LI R1,ATRIBU     ADRESSE DES DONNEES DES ATTRIBUTS DES LUTINS
LI R2,128        32 LUTINS
BLWP 2WMBW
B #R11
END CRAYON

```

Code machine à entrer en mémoire avec EASY BUG (option M)

```

7000 A5 5A 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7010 00 00 00 00 00 00 00 00 00 00 00 00 71 18 7F F8
7020 7E 22 A0 00 FF E0 20 00 3F FF 00 36 10 01 06 1C
7030 11 00 50 5C 00 04 10 0E 00 01 3A 20 20 20 20 20
7040 20 20 39 39 2F 34 20 41 53 20 20 20 20 20 20 20
7050 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
7060 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
7070 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
7080 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
7090 00 00 00 00 00 72 48 00 00 10 01 00 00 70 8F 00 0F

```

70A0	00	05	00	08	00	00	00	00	00	61	36	10	01	70	B8	7E	B8
70B0	90	00	70	08	62	FE	C0	00	1B	A8	72	44	00	04	00	00	00
70C0	00	00	72	E5	00	06	F8	00	00	00	00	01	00	00	7E	3E	3E
70D0	00	00	00	00	00	00	00	00	2D	0F	7F	F8	00	4F	00	3A	3A
70E0	7F	F8	A0	00	6F	06	7F	B8	00	3A	00	00	64	76	63	5A	5A
70F0	00	01	70	B8	7E	22	C0	00	00	00	00	00	00	00	00	00	00
7100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
7110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
7120	FF	FF	C7	B7	B7	B7	B7	B7	CB	FF	FF	C7	BB	BF	BB	C7	FF
7130	F3	F7	C7	B7	B7	B7	B7	B7	C3	FF	FF	C7	BB	83	BF	C7	FF
7140	E7	DB	DF	8F	DF	DF	DF	DF	9F	DF	DF	D7	CB	DB	DB	FF	FF
7150	EF	FF	8F	EF	EF	EF	83	FF	FF	FF	FF	57	AB	AB	AB	FF	FF
7160	FF	FF	87	DB	DB	C7	DF	8F	FF	FF	FF	A7	DB	DF	DF	FF	FF
7170	FF	FF	C7	BF	C7	BF	87	FF	FF	DF	DF	87	DF	DF	DF	E7	FF
7180	FF	FF	B8	B8	D7	D7	EF	FF	FF	FF	E7	D7	F7	F7	E3	FF	FF
7190	FF	E7	DB	FB	F7	EF	C3	FF	FF	FF	E3	FB	F7	FB	DB	E7	FF
71A0	FF	F7	E7	D7	D7	C3	F7	FF	FF	FF	E3	EF	E7	FB	DB	E7	FF
71B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
71C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
71D0	00	00	00	00	00	00	00	00	00	00	00	00	10	20	30	40	50
71E0	60	70	80	90	A0	B0	C0	D0	E0	F0	72	DA	71	EE	20	20	20
71F0	20	20	20	20	20	20	20	20	20	08	38	68	98	C8	F8	B7	B7
7200	AE	AA	A5	A5	A5	AA	AE	AE	AE	AE	C0	C0	00	22	20	21	01
7210	11	10	12	02	00	00	00	00	AE	1B	00	00	00	C0	1C	00	0F
7220	C0	1C	01	0F	C0	1C	02	0F	C0	18	03	0F	C0	13	04	0F	0F
7230	C0	13	05	0F	C0	13	06	0F	C0	18	07	0F	C0	18	0A	01	01
7240	C0	00	08	0F	C0	00	08	08	C0	00	09	0F	C0	F8	0B	0F	0F
7250	C0	F8	0B	0F	C0	F8	0B	0F	C0	F8	0B	0F	D0	00	20	30	30
7260	7C	7F	7F	3F	3F	1F	03	0F	3F	3F	1F	0F	03	00	1F	3F	3F
7270	3F	7F	7F	FC	30	00	FE	FC	7C	78	38	30	00	00	F8	FC	FC
7280	FC	FE	FE	3E	0C	04	C0	F0	F8	FC	FC	F0	C0	00	00	0C	0C
7290	3F	FE	FE	FC	FC	F8	00	00	18	38	3C	7C	7E	FE	10	10	10
72A0	00	C6	00	10	10	00	00	00	38	38	38	00	00	00	EE	FE	FE
72B0	FE	7C	FE	FE	EE	00	FF	FF	FF	FF	FF	FF	FF	FF	FF	00	00
72C0	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00	00
72D0	00	00	00	01	FF	FF	72	DA	7E	32	29	2E	2C	2F	23	30	30
72E0	22	31	24	32	2C	20	72	DA	72	FE	73	22	73	46	72	DA	DA
72F0	72	DA	74	92	74	FE	74	FE	74	FE	75	06	74	FE	2D	2E	2E
7300	27	2F	22	30	25	31	2A	32	2C	20	72	FE	72	FE	72	DA	DA
7310	72	FE	72	FE	72	DA	74	DE	74	F2	74	AC	74	B2	75	86	86
7320	74	FE	21	2E	2B	2F	26	30	2D	31	20	20	2C	20	73	22	22
7330	73	22	73	22	73	22	73	22	73	22	73	22	73	22	73	22	73
7340	75	72	75	06	74	FE	21	2E	24	2F	20	20	28	31	29	32	32
7350	2C	20	73	46	73	46	73	46	73	46	73	46	73	46	72	DA	75
7360	75	12	75	0A	75	18	75	40	74	FE	C8	08	72	D8	C8	20	20
7370	71	EA	72	D6	C8	20	71	EC	71	EA	06	A0	75	B6	C8	20	20
7380	72	D6	71	EA	D1	20	71	FF	D8	04	72	4C	D8	04	72	50	50
7390	D8	04	72	54	D8	04	72	58	02	00	1B	B0	02	01	72	4C	4C
73A0	02	04	03	E8	06	04	16	FE	02	04	00	14	04	C5	06	04	04
73B0	13	09	04	20	60	20	D1	60	83	75	02	85	12	00	16	F7	F7
73C0	04	60	75	9E	04	C6	04	C7	D1	E6	71	F9	D8	07	72	4D	4D
73D0	02	27	08	00	D8	07	72	51	02	27	08	00	D8	07	72	55	55
73E0	02	27	08	00	D8	07	72	59	04	20	60	28	02	04	00	18	18
73F0	04	20	60	20	D1	60	83	75	02	85	12	00	13	3C	02	85	85
7400	13	00	16	03	02	06	00	00	10	36	02	85	07	00	16	03	03
7410	02	06	00	01	10	30	02	85	08	00	16	03	02	06	00	02	02
7420	10	2A	02	85	09	00	16	03	02	06	00	03	10	24	02	85	85
7430	0A	00	16	03	02	06	00	04	10	1E	02	85	0B	00	16	03	03
7440	02	06	00	05	10	18	06	04	16	D3	05	86	02	86	00	05	05
7450	11	BB	13	04	06	A0	76	00	04	60	75	9E	02	07	F8	00	00
7460	D8	07	72	4D	D8	07	72	51	D8	07	72	55	D8	07	72	59	59
7470	04	20	60	28	10	BB	06	A0	76	00	C1	20	72	D6	02	24	24
7480	00	0C	0A	16	A1	06	C8	14	72	D6	02	24	00	0C	C0	54	54
7490	04	51	06	A0	7B	C4	02	E0	70	B8	10	31	06	A0	7C	DE	DE
74A0	06	A0	7C	FE	10	2C	06	A0	7D	22	10	29	06	A0	7B	6A	6A
74B0	10	26	C8	20	72	CA	72	CA	13	08	04	E0	72	CA	04	E0	E0
74C0	72	C0	C8	20	71	D6	71	D8	10	1A	05	A0	72	CA	05	A0	A0
74D0	72	C0	C8	20	71	D8	71	D6	06	A0	7B	6A	10	10	C8	20	20
74E0	72	C0	72	C0	16	03	05	A0	72	C0	10	09	04	E0	72	C0	C0
74F0	10	06	C8	20	72	C4	72	C4	16	03	05	A0	72	C4	10	4D	4D
7500	04	E0	72	C4	10	4A	06	A0	7D	FC	10	47	04	E0	72	CA	CA
7510	10	44	05	A0	72	CA	10	41	C8	20	72	C8	72	C8	13	0A	0A
7520	04	E0	72	CA	04	E0	72	C8	04	E0	72	C6	C8	20	71	D6	D6
7530	71	D8	10	33	05	A0	72	C8	C8	20	71	D8	71	D6	10	2D	2D
7540	C8	20	72	C8	72	C8	13	29	05	A0	72	C6	05	A0	72	C4	C4
7550	04	E0	72	C0	06	A0	7B	6A	06	A0	7D	22	10	1E	D0	60	60
7560	72	19	D8	20	72	D2	72	19	06	A0	7C	DE	D8	01	72	19	19
7570	10	14	D0	60	72	18	D8	20	72	0A	72	18	06	A0	7C	FE	FE
7580	D8	01	72	18	10	0A	C8	20	71	D8	71	D6	06	A0	7B	6A	6A
7590	06	A0	78	E8	C8	20	71	D6	71	D8	05	20	71	D0	C8	20	20
75A0	72	D6	71	EA	06	A0	75	B6	C2	E0	72	D8	02	01	80	00	00
75B0	06	01	16	FE	04	5B	C1	60	71	EA	02	09	00	02	02	04	04

75C0	00	08	02	00	17	08	02	00	08	02	01	71	18	04	C3			
75D0	D0	D5	06	C3	02	23	FF	E0	0A	33	A0	43	04	20	60	28		
75E0	05	85	06	09	13	07	02	00	08	06	04	16	EE	02	02			
75F0	00	10	04	58	02	09	00	02	02	20	00	28	06	04	10	E5		
7600	02	00	1B	80	02	01	72	4C	02	02	00	10	04	C4	D1	20		
7610	72	0A	D8	04	72	4C	D8	04	72	50	D8	04	72	54	D8	04		
7620	72	58	04	20	60	28	04	58	00	00	00	00	C8	08	76	28		
7630	C1	20	71	D0	02	84	12	00	13	60	04	C9	D8	20	72	05		
7640	72	44	D8	20	72	29	72	45	02	00	1B	A8	02	01	72	44		
7650	02	02	00	04	04	20	60	28	02	08	07	D0	02	07	02	58		
7660	04	20	60	20	D8	20	83	7C	83	7C	16	15	D1	20	83	75		
7670	02	84	FF	00	13	0C	02	89	00	03	11	05	02	07	00	C8		
7680	06	07	16	FE	10	09	06	07	16	EB	05	89	10	05	06	08		
7690	16	E7	04	60	77	BA	04	C9	D1	20	83	75	16	03	02	03		
76A0	00	07	10	28	02	84	0E	00	16	02	04	C3	10	23	02	84		
76B0	03	00	16	03	02	03	00	01	10	1D	02	84	06	00	16	03		
76C0	02	03	00	02	10	17	02	84	05	00	16	03	02	03	00	03		
76D0	10	11	02	84	04	00	16	03	02	03	00	04	10	08	02	84		
76E0	02	00	16	03	02	03	00	05	10	05	02	84	0F	00	16	04		
76F0	02	03	00	06	06	A0	77	C0	10	A1	02	02	00	04	02	0A		
7700	00	3F	04	CC	02	0D	00	04	D8	20	72	01	72	3C	02	00		
7710	1B	80	02	01	72	1C	04	C3	C1	C0	C2	01	02	09	00	14		
7720	02	00	1B	A0	02	01	72	3C	04	20	60	28	C0	07	C0	48		
7730	D4	63	72	00	04	20	60	28	04	C4	04	20	60	20	D1	20		
7740	83	75	02	84	12	00	13	23	06	09	16	F7	C0	07	C0	48		
7750	D4	60	72	0A	04	20	60	28	06	0A	13	22	C3	0C	13	10		
7760	04	CC	C0	C3	13	04	06	03	61	CD	62	0D	10	D7	02	00		
7770	1B	9C	02	01	72	38	02	03	00	07	C1	C0	C2	01	10	CE		
7780	02	83	00	07	13	C4	05	83	A1	CD	A2	0D	10	C7	06	A0		
7790	77	C0	02	0A	00	27	05	8C	02	09	00	32	04	60	77	20		
77A0	02	00	80	00	06	00	16	FE	D8	20	72	0A	72	3C	02	00		
77B0	1B	A0	02	01	72	3C	04	20	60	28	C2	E0	76	28	04	58		
77C0	C8	0B	76	2A	04	C0	D0	23	72	0C	C1	00	02	44	02	00		
77D0	16	0A	02	06	72	00	02	01	FF	FF	02	04	03	00	02	05		
77E0	00	01	06	A0	78	80	C1	00	02	44	01	00	16	0A	02	06		
77F0	72	00	02	01	00	01	02	04	B5	00	02	05	00	01	06	A0		
7800	78	80	C1	00	02	44	20	00	16	0A	02	06	72	19	02	01		
7810	FF	FF	02	04	0D	00	02	05	00	04	06	A0	78	80	C1	00		
7820	02	44	10	00	16	0A	02	06	72	19	02	01	00	01	02	04		
7830	FC	00	02	05	00	04	06	A0	78	80	D8	20	72	08	72	18		
7840	D8	20	72	01	72	3C	D8	20	72	29	72	3D	C8	00	71	D4		
7850	C8	01	71	D2	C8	20	72	C0	72	C0	72	C0	13	05	06	A0	7B	C4
7860	02	E0	70	B8	10	0A	C8	20	72	C6	72	C6	13	06	13	06	05	A0
7870	72	C2	06	A0	7D	22	04	E0	72	C2	C2	E0	76	2A	04	04	5B	
7880	91	16	13	07	02	04	00	09	06	C1	B5	81	A1	85	06	04		
7890	16	FC	04	5B	00	00	00	00	00	00	00	00	00	00	00	00	00	
78A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
78B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
78C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
78D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
78E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
78F0	C8	20	72	18	78	A8	B8	20	72	D3	78	A8	C8	20	72	D2		
7900	78	AA	02	03	00	04	02	23	FF	FC	11	78	04	C0	D0	23		
7910	78	A9	06	C0	04	C1	D0	63	78	A8	06	C1	C8	23	78	AA		
7920	78	9E	C8	00	78	A4	C8	00	78	A2	C2	80	06	A0	7A	68		
7930	13	65	06	A0	7A	06	02	81	00	04	11	E5	02	81	00	B4		
7940	15	E2	C8	20	72	D4	78	A0	88	20	78	A4	78	A6	13	18		
7950	88	20	78	A4	78	A6	15	08	C2	A0	78	A6	C3	20	78	A4		
7960	06	0C	06	A0	7A	F8	10	0F	C2	A0	78	A4	C3	20	78	A6		
7970	05	20	78	9E	A0	60	78	9E	06	0C	06	A0	7A	F8	05	20		
7980	78	9E	A0	60	78	9E	C8	20	78	A6	78	A4	A0	60	78	9E		
7990	C2	A0	78	A2	06	A0	7A	68	13	1F	C8	20	72	D2	78	A0		
79A0	02	0C	00	FD	05	20	78	9E	A2	A0	78	A0	83	0A	13	03		
79B0	06	A0	7A	68	16	F9	C3	20	78	A2	06	0A	C8	0A	78	A2		
79C0	05	8A	A0	60	78	9E	05	20	78	A0	06	A0	7A	F8	05	20		
79D0	78	9E	A0	60	78	9E	10	AD	C8	20	72	D4	78	A0	02	0C		
79E0	00	0D	A2	A0	78	A0	83	0A	13	03	06	A0	7A	68	13	F9		
79F0	88	0A	78	A4	11	88	C8	0A	78	A2	10	9B	02	E0	70	B8		
7A00	C2	E0	78	94	04	58	C8	0B	78	98	C0	20	78	A2	C0	80		
7A10	C2	80	06	A0	7A	68	06	A0	7A	9A	02	84	01	00	16	1E		
7A20	02	27	FF	F8	06	A0	78	36	02	82	00	10	11	13	C2	49		
7A30	16	11	02	0A	FF	00	06	A0	78	4C	D2	A0	71	D8	02	27		
7A40	20	00	06	A0	78	4C	02	27	E0	00	02	27	FF	F8	02	22		
7A50	FF	F8	10	E8	02	04	00	01	06	A0	7A	9A	05	82	C8	02		
7A60	78	A6	C2	E0	78	98	04	58	C8	0B	78	9C	C0	0A	C1	C1		
7A70	0A	57	E1	C1	02	47	FF	07	C1	00	02	44	00	07	A1	C0		
7A80	61	C4	06	A0	78	36	02	04	00	80	02	40	00	07	13	01		
7A90	08	04	C2	E0	78	9C	41	09	04	5B	41	09	13	0C	E2	44		
7AA0	06	02	02	82	00	0C	15	02	05	82	10	05	0A	14	02	84		
7AB0	01	00	13	01	10	F2	C8	0B	78	96	02	27	20	00	06	C7	D8	07
7AC0	7B	4C	10	02	C8	0B	78	96	02	27	20	00	06	C7	D8	07		
7AD0	8C	02	06	C7	D8	07	8C	02	D2	A0	71	D8	D3	60	88	00		

7AE0	02 4A F0 00 02 4D 0F 00 F2 8D 06 A0 7B 4C 02 27	7D70	72 18 72 16 04 E0 72 C2 D2 60 72 15 09 89 D1 E0
7AF0	E0 00 C2 E0 78 96 04 5B C8 0B 78 9A A2 A0 78 A0	7D80	72 17 09 87 C8 07 7B C0 D2 A0 72 14 09 8A 05 8A
7B00	83 0A 13 16 06 A0 7A 68 13 F9 C8 CA 78 A8 06 C1	7D90	D2 20 72 16 09 88 05 88 C8 08 7B C2 04 CC C8 20
7B10	D8 C1 78 A8 06 C1 C8 E0 78 9E 78 AA 02 23 00 04	7DA0	72 D2 7B BC C8 20 72 D2 7B BE 61 C9 11 01 10 04
7B20	A2 A0 78 A0 83 0A 13 04 06 A0 7A 68 16 F9 10 E6	7DB0	C8 20 72 D4 7B BC 05 07 C1 C7 16 01 07 0C 62 0A
7B30	C2 E0 78 9A 04 5B 06 C7 D8 07 8C 02 06 C7 D8 07	7DC0	11 01 10 04 C8 20 72 D4 7B BE 05 08 C0 09 C0 4A
7B40	8C 02 04 C9 D2 60 88 00 06 C9 04 5B C2 07 02 0C	7DD0	06 01 06 A0 7C 20 88 09 7B C0 16 03 88 0A 7B C2
7B50	40 00 06 C8 D8 08 8C 02 06 C8 E2 0C D8 08 8C 02	7DE0	13 0A C3 0C 11 04 A2 60 7B BC 63 08 10 EF A2 A0
7B60	10 00 D8 0A 8C 00 04 5B 00 00 C8 0B 78 68 04 C3	7DF0	7B BE A3 07 10 EB 02 E0 70 B8 04 5B 02 00 20 00
7B70	D8 03 83 74 04 20 60 20 D0 E0 83 7C 13 F8 D0 E0	7E00	02 01 11 00 04 20 60 24 05 80 02 80 37 00 16 FA
7B80	83 75 0B 83 02 83 00 30 11 F2 02 83 00 39 12 09	7E10	04 C0 04 C1 04 20 60 24 05 80 02 80 17 00 16 FA
7B90	02 83 00 41 11 EC 02 83 00 46 15 E9 02 23 FF C9	7E20	04 5B 02 E0 70 E8 06 A0 7E E8 06 A0 7F 8E 06 A0
7BA0	10 02 02 23 FF D0 D8 23 71 DA 71 D8 02 03 01 00	7E30	73 6A 04 E0 72 BE 06 A0 7E BA 06 A0 7E 88 03 00
7BB0	D8 03 83 74 C2 E0 7B 68 04 5B 00 00 00 00 00	7E40	00 02 03 00 00 00 05 A0 72 BE 06 A0 7E BA 02 01
7BC0	00 00 00 00 C8 0B 7B BA 02 E0 71 B0 C2 E0 7B BA	7E50	00 64 06 01 13 EE 04 20 60 20 04 C0 D0 20 83 75
7BD0	D0 20 72 19 09 80 D0 60 72 18 09 81 10 21 C8 0B	7E60	02 80 FF 00 13 F6 C8 00 71 D0 06 A0 73 6A C8 20
7BE0	7B BA 02 E0 71 B0 C2 E0 7B BA C0 20 71 D4 C0 60	7E70	71 D0 71 D0 11 E4 04 E0 72 BE 06 A0 7E BA 06 A0
7BF0	71 D2 02 80 00 0D 11 72 02 80 00 FC 15 6F 02 81	7E80	7E 88 06 A0 76 2C 10 D5 D8 20 72 05 72 44 D8 20
7C00	00 03 11 6C 02 81 00 B5 15 69 10 0A C8 0B 7B BA	7E90	72 29 72 45 02 00 1B A8 02 01 72 44 02 02 00 04
7C10	02 E0 71 B0 C2 E0 7B BA C0 20 71 D4 C0 60 71 D2	7EA0	04 20 60 28 02 03 4E 20 06 03 10 00 16 FD C8 20
7C20	C1 01 0A 54 E1 01 02 44 FF 07 C1 40 02 45 00 07	7EB0	72 0A 72 44 04 20 60 28 04 5B C8 20 72 BE 72 BE
7C30	A1 00 61 05 C8 20 72 CA 13 15 02 05 40 00	7EC0	13 07 D8 20 72 05 72 48 D8 20 72 29 72 49 10 03
7C40	02 24 20 00 06 C4 D8 04 8C 02 06 C4 D8 04 8C 02	7ED0	C8 20 72 0A 72 48 02 00 1B AC 02 01 72 48 02 02
7C50	D0 20 71 D8 D1 A0 88 00 02 46 F0 00 02 40 F0 00	7EE0	00 04 04 20 60 28 04 5B 02 00 01 F0 04 20 60 34
7C60	09 40 F1 80 10 31 06 C4 D8 04 8C 02 06 C4 D8 04	7EF0	02 00 1C 00 04 C2 04 C1 04 C3 04 20 60 24 05 80
7C70	8C 02 C0 05 D1 A0 88 00 02 03 80 00 0B 03 02 05	7F00	05 83 02 21 01 00 02 83 01 00 16 F7 05 82 02 82
7C80	40 00 C8 20 72 C4 72 C4 16 02 51 83 10 1D F1 83	7F10	00 03 16 F1 04 C0 04 C1 04 20 60 24 05 80 02 80
7C90	C0 04 02 20 20 00 06 C0 D8 00 8C 02 06 C0 D8 00	7F20	18 00 16 FA 02 00 20 00 02 01 11 00 04 20 60 24
7CA0	8C 02 D0 60 71 D8 D0 E0 88 00 02 41 F0 00 02 43	7F30	05 80 02 80 37 00 16 FA 02 00 37 00 02 01 F1 00
7CB0	0F 00 F0 43 06 C0 D8 00 8C 02 06 C0 E0 05 D8 00	7F40	04 20 60 24 05 80 02 80 38 00 16 FA 02 00 00 02
7CC0	8C 02 10 00 D8 01 8C 00 06 C4 D8 04 8C 02 06 C4	7F50	04 20 60 34 02 00 01 E0 04 20 60 34 02 00 02 07
7CD0	E1 05 D8 04 8C 02 10 00 D8 06 8C 00 04 5B 02 E0	7F60	04 20 60 34 02 00 03 FF 04 20 60 34 02 00 04 03
7CE0	71 B0 D2 60 72 18 09 89 02 08 00 0D C0 08 C0 49	7F70	04 20 60 34 02 00 05 37 04 20 60 34 02 00 06 03
7CF0	06 A0 7C 20 05 88 02 88 00 FD 11 F8 10 0F 02 E0	7F80	04 20 60 34 02 00 07 01 04 20 60 34 04 5B 02 03
7D00	71 B0 D2 20 72 19 09 88 02 09 00 03 C0 08 C0 49	7F90	01 00 D8 03 83 74 02 00 18 00 02 01 72 5E 02 02
7D10	06 A0 7C 20 05 89 02 89 00 B6 16 F8 02 E0 70 B8	7FA0	00 60 04 20 60 28 02 00 1B 80 02 01 72 1C 02 02
7D20	04 5B C8 20 72 C2 72 C2 16 15 C8 20 72 18 72 14	7FB0	00 40 04 20 60 28 04 5B 00 00 00 00 00 00 00
7D30	C8 20 72 D2 72 C2 D8 20 72 05 72 40 D8 20 72 29	7FC0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7D40	72 41 02 00 1B A4 02 01 72 40 02 00 04 04 20	7FD0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7D50	60 28 04 5B 02 E0 71 B0 C8 20 72 0A 72 40 02 00	7FE0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7D60	1B A4 02 01 72 40 02 02 00 04 04 20 60 28 C8 20	7FF0	00 00 00 00 00 00 00 00 43 52 41 59 4F 4E 7E 22

Le traitement des fichiers en PASCAL

Gérard Santraille

Les adversaires du Pascal argueraient, à juste titre d'ailleurs, que ce langage pêche par sa lourdeur au niveau des manipulations de fichiers. La première version de Pascal (proposée par N. Wirth dans les années 1970) était pratiquement dépourvue d'instructions permettant de gérer efficacement les entrées/sorties sur fichiers.

Depuis ces temps héroïques, Pascal s'est considérablement étoffé et tous les compilateurs offrent désormais de nombreuses commandes permettant l'accès à de très nombreux types de fichiers : nous verrons par la suite ceux qui sont prédéfinis. Quant aux méthodes d'accès, elles se répartissent très classiquement en deux catégories :

- **accès séquentiel** : les données sont lues et/ou écrites les unes après les autres.

- **accès direct** : on se positionne directement sur un enregistrement pour lire ou écrire en spécifiant sa position par rapport au début du fichier.

Toutes les autres techniques, plus ou moins élaborées, d'accès aux fichiers comme le *séquentiel indexé* ou le *Hash-codings* s'appuient sur ces deux types fondamentaux.

Notions de bases :

Fichiers
Enregistrements
Tampons

Un fichier est un ensemble (de données) structuré en enregistrements. On symbolise souvent un fichier par une feuille de papier et ses enregistrements par les lignes formant le texte (contenu du fichier).

Prenons un exemple concret. Supposons que l'on désire collecter des informations sur un grand nombre de personnes (fichier de relations). On dispose pour chaque individu de son

nom, de son prénom et de son adresse. Manuellement on écrirait sur une feuille, pour chaque personne, la liste des informations connues.

La déclaration Pascal correspondant à une personne pourrait être :

type

personne = packed record of

```
nom      : string[20];
prenom   : string[15];
tel_dom  : string[13];
tel_bur  : string[22];
adr1     : string[25];
adr2     : string[20];
adr3     : string[10];
code     : string[6];
ville    : string[20];
end;
```

Cet ensemble d'informations, relatif à une seule personne, correspond à un enregistrement (**record**).

A l'attention des puristes signalons que le terme **Enregistrement** est ici, propre au Pascal. Il ne correspond pas à la notion "d'enregistrement de fichier" communément admise par les informaticiens.

Dans notre cas l'enregistrement est une entité dont l'organisation (le **type**) est entièrement définissable par l'utilisateur. Ce dernier est donc libre de créer la structure qui correspondra le mieux à son problème. La taille et la complexité d'un 'record' n'ont pour limite que l'imagination du programmeur.

L'assemblage de tous les enregistrements va former un fichier. Un fichier est donc une suite d'enregistrements qui :

- ont tous la même structure (le même type)
- peuvent eux-même être composés de plusieurs éléments (appelés **champs**).

La déclaration d'un tel fichier se fait dans la rubrique **Var** par :

fichier : file of **personne**;

Notons que la définition du type fichier ne précise que le type des éléments qui le compose.

Remarque : Un enregistrement peut lui-même être composé d'enregistrements. On peut ainsi bâtir des structures imbriquées extrêmement complexes. La composition des données en ensembles puis en ensembles d'ensembles est l'une des grandes originalités du Pascal.

La déclaration d'une variable fichier **F** introduit automatiquement une variable associée au fichier. Cette variable s'appelle le **tampon** (ou la **fenêtre**) de fichier et son type est celui des composants du fichier. Littéralement le fichier est désigné par **F**.

Si nous reprenons notre feuille de papier pour représenter un fichier, la fenêtre correspondrait à une règle que l'on déplacerait verticalement de ligne en ligne. Dans cette représentation chaque ligne équivaut à un enregistrement. A un instant donné, on ne peut accéder pour lire ou écrire qu'à l'enregistrement situé dans la fenêtre (la ligne pointée par la règle). Cet enregistrement est appelé **élément courant**. Une opération de lecture (ou d'écriture) sur le fichier à pour action de déplacer l'élément courant dans le tampon du fichier.

Toutes les actions d'entrées/sorties se font par l'intermédiaire de ce tampon. Ce dernier se comporte comme une variable du type des enregistrements du fichier. Les accès à un fichier (sauf comme nous le verrons plus loin pour certains fichiers de type prédéfinis) se décomposent de la façon suivante :

Lecture

- positionnement du tampon à l'endroit souhaité

Ecriture

- positionnement du tampon à l'endroit souhaité

pour l'accès direct seulement

- lecture du fichier vers la variable tampon

- écriture (assignation) de la variable tampon

- visualisation de la variable tampon

- écriture physique sur le fichier

Instructions élémentaires

Lecture

Reprenons notre fichier **F** : file of personne avec :

personne = **packed record of**

```
nom      : string[20];
prenom   : string[15];
tel_dom  : string[13];
tel_bur  : string[22];
adr1     : string[25];
adr2     : string[20];
adr3     : string[10];
code     : string[6];
ville    : string[20];
end;
```

L'ouverture d'un fichier en lecture se fait au moyen de l'instruction **RESET(F, 'nom_du_fichier')** ou **RESET(F)** qui est équivalent à **RESET(F, 'F')**. Cette opération positionne la fenêtre sur le premier élément du fichier et le tampon **F** est initialisé avec la valeur de l'élément courant placé sous la fenêtre.

Lors de l'ouverture par **RESET** le premier élément est automatiquement lu. La variable tampon contient donc le premier enregistrement : il n'est pas nécessaire de le lire.

On peut manipuler les différents éléments de cet enregistrement de la même façon qu'une variable classique du même type. Ainsi si l'on désire visualiser à l'écran le champ de l'enregistrement contenant le nom de la personne il suffit d'écrire **WRITELN(F.NOM)**.

Pour accéder à l'enregistrement suivant du fichier il faut exécuter la commande **GET(F)**. Cette instruction a deux effets :

- la fenêtre de lecture avance d'un enregistrement
- la variable **F** prend les nouvelles valeurs de l'enregistrement pointé par la fenêtre.

la lecture d'un enregistrement se fait donc en deux étapes.

- transfert des informations de l'enregistrement vers la variable tampon **F**
- visualisation de cette variable à l'écran.

On peut de cette façon parcourir *séquentiellement* tout le fichier et l'afficher à l'écran. Lorsque la fin du fichier est atteinte la fonction intrinsèque **EOF(F)** devient vraie.

La procédure suivante permet de lire en entier notre fichier et d'afficher les noms de chaque enregistrement.

procédure LireFichier (var fichier : file of personne);

```
begin
  reset(f, fichier);
  while not eof(f) do
    begin
      writeln(f.nom);
      get(f);
    end;
  close(f, lock);
end.
```

Notons la présence du **while not eof(f)** qui permet d'effectuer une séquence d'instructions tant que **eof(f)** est *faux* donc tant que la fin du fichier n'a pas été atteinte.

Remarque : L'ouverture d'un fichier au moyen de **RESET** permet également de le mettre à jour et donc de modifier certains enregistrements ou d'en créer de nouveaux. Par contre **RESET** ne permet pas de créer un fichier : il faut que le fichier soit déjà physiquement

présent sur la disquette pour pouvoir être ouvert par **RESET**.

Ecriture

Comme pour la lecture, la première opération doit être l'ouverture du fichier. Cette action associe, comme nous l'avons vu, un tampon au fichier qui sera utilisé pour les opérations d'entrées/sorties.

L'écriture sur un fichier se fera en deux étapes.

- écriture dans la variable tampon **F** (affectation)
- écriture de **F** sur le fichier.

L'instruction qui permet d'ouvrir un fichier en écriture est :

REWRITE(F, 'nom_du_fichier') ou **REWRITE(F)** qui est équivalent à **REWRITE(F, 'F')**.

Celle qui écrit la variable tampon **F** sur le fichier est **PUT(F)**. Par ailleurs, **REWRITE** est le mode d'ouverture qui doit être utilisé pour créer un nouveau fichier.

Ainsi si l'on désire écrire sur le fichier l'enregistrement correspondant à une personne il convient de définir chacun des éléments de **F** puis d'écrire la variable.

Exemple :

```
.....
f.nom      := 'Jackson';
f.prenom   := 'Michael';
f.tel_dom  := '(43) 66.77.88';
f.tel_bur  := '(8) 22.11.22';
f.adr1     := '28, Melody Street';
f.adr2     := 'flat 23WZ, 1024th
floor';
f.adr3     := '';
f.code     := '43.122';
f.ville    := 'THRILLER CITY';

.....

put(f);
.....
```

Les deux instructions **PUT/GET** constituent les deux éléments de base pour pouvoir dialoguer avec un fichier. Nous venons de voir comment ouvrir, lire et écrire dans un fichier. Une fois le travail terminé il convient de le fermer. La fermeture d'un fichier provoque plusieurs actions :

- dissociation de la variable **F** associée à 'nom_du_fichier' lors du **Reset(F, 'nom_du_fichier')**.
- restitution au système de la mémoire utilisée par la variable tampon **F**.

C'est l'instruction **CLOSE(F,option)** qui a cette lourde tâche. Le mot-clé option permet d'effectuer divers types de fermeture.

LOCK : c'est l'option que l'on emploie dans la majorité des cas. Le fichier est fermé et toutes les opérations sauvegardées.

PURGE : à manipuler avec beaucoup de précautions. Le fichier est détruit qu'il ait été ouvert en lecture ou en écriture.

CRUNCH : le fichier est fermé et tronqué à l'endroit où se trouvait la fenêtre. Toutes les informations situées au delà sont perdues.

NORMAL : si le fichier a été ouvert en lecture il est fermé et aucune action particulière n'a lieu. S'il a été ouvert en écriture il est détruit comme avec **PURGE**.

Si le mot-clé option est absent l'option par défaut est **NORMAL**.

Différents modes d'accès

Toutes les opérations que nous venons de décrire utilisent l'accès séquentiel (le plus simple mais bien souvent le moins performant). Il existe de nombreuses techniques permettant d'isoler une information noyée dans une masse importante.

Si l'on cherche la signification du mot **Seringa** dans le dictionnaire (plante à fleurs blanches très odorantes) il est bien évident que l'on ne va pas parcourir l'ouvrage de la première page à celle contenant notre cible. On effectue une recherche par encadrements successifs du mot à trouver. Cette technique qui s'apparente à la dichotomie est évidemment beaucoup plus efficace.

Une première remarque s'impose. Une recherche autre que séquentielle ne peut se faire que sous certaines conditions. Par exemple on pressent bien qu'une recherche par encadrements successifs ne peut se faire que si l'ensemble de travail est ordonné.

Plus généralement, toutes les méthodes d'accès à des informations contenues dans un fichier imposent

certaines restrictions et ne peuvent s'appliquer à tous les types d'ensembles.

Pour pouvoir gagner du temps lors de l'accès il faut inévitablement faire l'une (ou les deux) de ces concessions :

- Fichier à structure particulière (comme le dictionnaire). Le domaine sur lequel va s'effectuer la recherche doit présenter certaines caractéristiques particulières. Il sera partiellement ou totalement ordonné ou encore ses éléments seront regroupés par famille suivant un critère de sélection.
- Stockage d'informations supplémentaires.

Reprenons l'exemple de la recherche d'un mot dans une ensemble de taille respectable (toute la langue française s'il s'agit du dictionnaire) et voyons sur ce cas précis, facilement généralisable, les principales méthodes.

Recherche séquentielle

Aucune contrainte particulière : l'ensemble peut être quelconque et il n'est pas nécessaire de stocker d'autres informations que les mots eux-mêmes. Ne peut s'appliquer qu'à des ensembles de petite taille.

Recherche séquentielle indexée

Comme son nom l'indique cette technique fait appel à un index. On répartit les éléments de l'ensemble (les mots du dictionnaire) en plusieurs classes ou sous-ensembles. Chaque classe contiendra les mots commençant par la même lettre. On note ensuite dans un index les positions (par rapport au début du fichier) de chaque classe. Pour rechercher un mot on se positionne dans la classe correspondante (on connaît toutes les positions) et on entreprend dans la classe une recherche séquentielle. Cette technique permet donc de partitionner le domaine de recherche pour travailler sur un ensemble beaucoup plus petit.

Les contraintes sont ici de deux types :

- l'ensemble doit être découpé en plusieurs sous-ensembles. Il donc nécessaire de lui appliquer un

certain traitement. Notons qu'au sein d'une classe les éléments ne sont pas ordonnés. De même les classes étant complètement indépendantes, elles peuvent apparaître dans n'importe quel ordre

- il est nécessaire de connaître les emplacements de toutes les classes donc de stocker une information supplémentaire.

Il est parfaitement possible d'augmenter le nombre de classes afin de réduire encore le temps d'accès. Si l'on prend en compte les deux premières lettres on passe de 26 sous-ensembles à 676 (26^2). Les classes devenant de plus en plus petites les temps d'accès diminuent mais la quantité d'informations à stocker augmente.

On peut ainsi faire une partition de plus en plus fine et même travailler sur un ensemble dont chaque élément constitue une classe. L'accès à l'information est alors instantané : on effectue de l'accès direct pur.

Remarquons que dans ce dernier cas il ne reste plus aucune contrainte sur l'ensemble puisque le travail de regroupement des éléments en classes est immédiat. Un singleton est en effet toujours 'regroupé' avec lui-même. Ce découpage présente donc un intérêt particulier.

Le nombre de positions à mémoriser étant très important (un entier par élément) il serait extrêmement intéressant de définir une fonction permettant à partir de l'élément de *calculer* sa position. Il ne serait alors plus nécessaire de mémoriser toutes ces positions. Cette technique s'appelle le **Hash-coding**.

Le Hash-coding

Sans trop entrer dans le détail décrivons le principe de cette méthode simple et pourtant peu connue. L'idée consiste donc à calculer la position de l'élément dans le fichier en lui appliquant une certaine fonction (par exemple la somme des positions dans l'alphabet des lettres composant le mot). Cette technique suppose que l'on dispose d'une fonction suffisamment sélective pour pouvoir qualifier différemment et par là même pouvoir différencier chaque élément.

Ce n'est que très rarement le cas et plusieurs éléments peuvent alors fournir la même adresse de rangement, on parle de collisions. Il est malheureusement impossible de s'affranchir de cette limitation. On la contourne souvent en juxtaposant les éléments fournissant la même adresse. La recherche se fait alors en deux temps :

- calcul de la valeur à partir de l'élément et positionnement en accès direct sur l'enregistrement adéquat
- recherche séquentielle à partir de ce point d'entrée

Cette technique s'apparente donc fortement au séquentiel indexé. Bien qu'elle semble plus intéressante (rappelons qu'aucune information supplémentaire n'est à conserver), elle souffre, de par sa conception, d'une très importante restriction.

L'ensemble des adresses générées par la fonction n'est pratiquement jamais continu : c'est à dire qu'il peut comporter des trous. Ainsi, si l'ensemble des adresses est {2,11,3,20,7} on va utiliser 20 enregistrements (l'adresse la plus élevée) pour ranger 5 éléments. Certains systèmes d'exploitation réservent la place physique d'un enregistrement que s'il est non-vide. Cette discontinuité n'a alors aucune importance. Ce n'est hélas pas le cas de notre TI-99 qui réserve la place nécessaire à n enregistrements si on accède au $n^{\text{ième}}$ même s'ils sont tous vides....

L'accès direct en Pascal

Nous réalisons que toutes ces techniques utilisent abondamment le positionnement direct sur un enregistrement. L'instruction Pascal qui permet cela est **SEEK(F,position)**. La position du premier enregistrement d'un fichier est zéro.

L'exécution de cette instruction provoque le déplacement de la fenêtre sur l'enregistrement spécifié. Il convient ensuite de faire **GET(F)** ou **PUT(F)** selon que l'on désire lire ou écrire.

Avec cette dernière commande nous disposons d'une panoplie d'instructions permettant de créer et mettre à jour les fichiers de tous poils.

Fichiers de types prédéfinis

Il existe deux principaux types de fichiers pré-établis. Ces fichiers peuvent être manipulés par d'autres instructions. Ils sont plus faciles à mettre à œuvre mais admettent certaines restrictions. Il s'agit des types **TEXT** et **INTERACTIVE**. Tous les deux sont composés de caractères. Les fichiers de ce type n'ont pas à être définis avec le mot-clé **file** mais simplement dans la rubrique **Var** comme une variable classique.

Ainsi

```
var f : text;
Indique au programme que f est un
fichier de caractères exactement
comme
```

```
var f : file of char;
l'aurait fait.
```

Les fichiers de caractères ont une structure particulière. Ils sont formés d'enregistrements (eux-mêmes composés de caractères) de taille différente. Chaque enregistrement (ligne de texte) se termine par un caractère spécial qui marque la fin de la ligne. La présence de cette marque peut être testée par la fonction booléenne **EOLN(F)** qui est vraie si la fin de la ligne a été atteinte et fausse autrement.

La lecture (respectivement l'écriture) sur un tel fichier se fait par **READ(F,variable_lue)** et **READLN** (respectivement **WRITE** ou **WRITELN**). Ces instructions ont pour premier argument la variable fichier (celle utilisée lors du **RESET** ou du **REWRITE**) suivi de la (ou des) variable(s) à lire (ou à écrire). **READ (WRITE)** ne lit (écrit) qu'un seul caractère alors que **READLN (WRITELN)** manipule une ligne entière.

Si **F** est de type **TEXT** et **CH** de type **CHAR** **READ(F,CH)** lit le caractère situé sous la fenêtre. Cette instruction est donc équivalente à

```
CH=F^;
```

```
GET(F);
```

alors que pour un fichier de type **INTERACTIVE** l'ordre de ces instructions est inversé ainsi **READ(F,CH)** est équivalent à

```
GET(F);
```

```
CH=F^;
```

Cette différence peut paraître minime, pourtant elle a son importance. Par exemple si on teste la fin de la ligne au moyen de **EOLN(F)**, pour un fichier de type **INTERACTIVE** **EOLN** ne devient vraie qu'après avoir lu le caractère qui marque la fin de la ligne.

Cette marque est donc lue avec un fichier **INTERACTIVE**, ce n'est pas le cas avec un fichier **TEXT**.

Les fichiers créés à partir de l'éditeur de texte du système peuvent être lus (et modifiés) en les considérant comme **TEXT** ou **INTERACTIVE**.

La principale limitation des ces fichiers est inhérente à la structure même de leurs enregistrements. Comme ils ont une taille variable (la marque de fin de ligne peut se trouver n'importe où) le système est incapable de calculer l'emplacement physique d'un enregistrement donné sans parcourir le fichier jusqu'à lui. L'accès direct n'est donc pas envisageable. Il est néanmoins possible de faire de l'accès direct sur un fichier de caractères en déclarant explicitement sa structure, les enregistrements auront alors une taille fixe. Ainsi :

```
type blabla = packed record of
    ligne : string[80]
end;
```

```
var fichier : file of blabla;
```

ou directement :

```
var fichier : file of string[80];
permet d'utiliser l'instruction SEEK.
```

Programme exemple

Le petit programme suivant illustre certaines des commandes décrites auparavant. Le fichier **ADRESSES.DATA** regroupe les informations de toutes les personnes.

La technique employée ici est l'accès direct simple pour chaque enregistrement. En début d'exécution ce fichier est intégralement parcouru et un *tableau-index* est créé. Ce tableau regroupe les cinq premières lettres du champ 'nom' de chaque enregistrement du fichier.

Ainsi pour accéder aux informations relatives à une personne il faut donner au programme son nom. Il cherche alors dans l'index la position de l'enregistrement correspondant à cette personne.

Notez dans le listing la présence de la directive `[$!-]` qui, associée à `IORESULT`, permet de tester la présence de `ADRESSES.DAT`. Si ce fichier est absent, `RESET` provoque une erreur : on emploie alors `REWRITE`. On

crée ensuite le premier enregistrement du fichier, on le ferme en le sauvegardant, puis on l'ouvre à nouveau avec `RESET` qui permet aussi bien de lire que d'écrire.

De conception très simple ce programme gère directement le fichier sur disquette. Les opérations de lecture ou d'écriture sont évidemment plus longues (l'écriture sur fichier est beaucoup moins rapide que

l'affectation d'une variable) mais le programme peut ainsi appréhender des quantités d'informations qui ne "tiendraient" pas en mémoire centrale.

A vous d'enrichir ce programme en lui adjoignant, par exemple, des algorithmes de tri, d'édition sélective ou de sélection multi-critères. Faites nous part de vos suggestions...



```
program FICHES;

const bidon=false;

var f      : file of record
    nom    : string[20];
    prenom : string[15];
    tel_dom : string[13];
    tel_bur : string[22];
    adr1    : string[25];
    adr2    : string[20];
    adr3    : string[10];
    code    : string[6];
    ville   : string[20];
end;

    choix,nmax : integer;
    pers       : packed array[1..50] of string[5];
```

```
procedure poursuivre;
begin
    gotoxy(15,23);
    write('ENTER pour continuer',chr(7));
    readln
end;
```

```
procedure lister;
var i : integer;
begin
    page(output);
    writeln('L I S T E   D E S   N O M S');
    writeln('-----');
    gotoxy(0,5);
    seek(f,0);
    for i:=1 to nmax do
        begin
            get(f);
            writeln('':15,f^.nom)
        end;
    for i:=1 to 6 do writeln;
    poursuivre
end;
```

```
procedure consulter;
var i : integer;
    nom1: string[5];
    nom2: string[20];
begin
    page(output);
    writeln('C O N S U L T E R   U N E   F I C H E');
    writeln('-----');
    gotoxy(0,8);
    write('Nom      : ');
    readln(nom2);
    if length(nom2)<5 then
        nom2:=concat(nom2,' ');
    nom1:=copy(nom2,1,5);
    i:=0;
```

```
repeat
    i:=i+1
until (pers[i]=nom1) or (i=nmax+1);
if i=nmax+1 then
    begin
        gotoxy(15,19);
        writeln('non repertorie ...');
    end
else
    begin
        seek(f,i-1);
        get(f);
        writeln('':10,f^.prenom);
        writeln('Tel dom : ',f^.tel_dom);
        writeln('Tel bur : ',f^.tel_bur);
        writeln('':10,f^.adr1);
        writeln('':10,f^.adr2);
        writeln('':10,f^.adr3);
        writeln('':10,f^.code,'':13,f^.ville);
    end;
poursuivre
end;
```

```
procedure modifier;
var nom1: string[5];
    nom2: string[20];
    i : integer;
begin
    page(output);
    writeln('M O D I F I E R   U N E   F I C H E');
    writeln('-----');
    gotoxy(0,5);
    write('Nom      : ');
    readln(nom2);
    if length(nom2)<5 then
        nom2:=concat(nom2,' ');
    nom1:=copy(nom2,1,5);
    i:=0;
    repeat
        i:=i+1
    until (pers[i]=nom1) or (i=nmax+1);
    if i=nmax+1 then
        begin
            gotoxy(15,19);
            writeln('non repertorie ');
        end
    else
        begin
            seek(f,i-1);
            get(f);
            writeln('':10,f^.prenom);
            writeln('Tel bur : ',f^.tel_bur);
            write('nou.val. : ');
            readln(f^.tel_bur);
            writeln('Tel dom : ',f^.tel_dom);
            write('nou.val. : ');
            readln(f^.tel_dom);
            writeln('Adr1    : ',f^.adr1);
            write('nou.val. : ');
            readln(f^.adr1);
```

```

writeln('Adr2      : ',f^.adr2);
write('nou.val.: ');
readln(f^.adr2);
writeln('Adr3      : ',f^.adr3);
write('nou.val.: ');
readln(f^.adr3);
writeln('Code      : ',f^.code);
write('nou.val.: ');
readln(f^.code);
writeln('Ville     : ',f^.ville);
write('nou.val.: ');
readln(f^.ville);
seek(f,i-1);
put(f);
end;
poursuivre
end;

```

```

procedure supprimer;
var nom1: string[5];
    nom2: string[20];
    i,j : integer;
begin
  page(output);
  writeln('S U P P R I M E R   U N E   F I C H E');
  writeln('-----');
  gotoxy(0,8);
  write('Nom      : ');
  readln(nom2);
  if length(nom2)<5 then
    nom2:=concat(nom2,' ');
  nom1:=copy(nom2,1,5);
  i:=0;
  repeat
    i:=i+1
  until (pers[i]=nom1) or (i=nmax+1);
  if i=nmax+1 then
    begin
      gotoxy(15,19);
      writeln('non repertorie ...');
    end
  else
    begin
      for j:=i+1 to nmax do
        begin
          pers[j-1]:=pers[j];
          seek(f,j-1);
          get(f);
          seek(f,j-2);
          put(f);
        end;
      nmax:=nmax-1;
    end;
  poursuivre
end;

```

```

procedure ajouter;
begin
  page(output);
  writeln('A J O U T E R   U N E   F I C H E');
  writeln('-----');

  seek(f,nmax);
  gotoxy(0,8);
  write('Nom      : '); readln(f^.nom);
  if length(f^.nom)<5 then
    f^.nom:=concat(f^.nom,' ');
  write('Prenom : '); readln(f^.prenom);
  write('Tel dom : '); readln(f^.tel_dom);
  write('Tel bur : '); readln(f^.tel_bur);
  write('Adr. 1/3: '); readln(f^.adr1);
  write('Adr. 2/3: '); readln(f^.adr2);
  write('Adr. 3/3: '); readln(f^.adr3);
  write('Code   : '); readln(f^.code);
  write('Ville  : '); readln(f^.ville);

```

```

nmax:=nmax+1;
pers[nmax]:=copy(f^.nom,1,5);
put(f);
end;

procedure logo;
begin
  page(output);
  writeln('15, Gestion d''un fichier d''adresses');
  writeln('(C)opyright 99 Magazine et G. SANTRAILLE');
  gotoxy(10,10);
end;

```

```

begin      (debut du programme principal)
($I-)
reset(f,'ADRESSES.DAT');
($I+)
if ioresult<>0 then
  begin
    page(output);
    nmax:=0;
    writeln('ADRESSES.DAT n''existe pas');
    writeln('definition du premier enregistrement');
    rewrite(f,'ADRESSES.DAT');
    poursuivre;
    ajouter;
    close(f,lock);
    reset(f,'ADRESSES.DAT');
  end;
logo;
gotoxy(15,10);
writeln('Un peu de patience');

```

```

    nmax:=0;
    while not eof(f) do
      begin
        nmax:=nmax+1;
        pers[nmax]:=copy(f^.nom,1,5);
        get(f);
      end;

  repeat
    logo;
    writeln('1 -> Liste des noms');
    writeln('10,2 -> Consulter une fiche');
    writeln('10,3 -> Modifier une fiche');
    writeln('10,4 -> Supprimer une fiche');
    writeln('10,5 -> Ajouter une fiche');
    writeln('10,6 -> Finir');
    writeln;
    writeln('20, Votre choix ?');
    repeat
      writeln(chr(7));
      gotoxy(34,17);
      readln(choix);
    until (choix<7) and (choix>0);
    case choix of
      1 : lister;
      2 : consulter;
      3 : modifier;
      4 : supprimer;
      5 : ajouter;
      6 : begin
          page(output);
          seek(f,nmax);
          close(f,crunch);
          exit(program)
        end
    end;
  until bidon=true
end.

```


Editeur de caractères et lutins

Julien Thomas

Avec ce programme, écrit en Basic Etendu, il est possible de se passer de l'éternelle feuille de papier quadrillée pour obtenir les codes nécessaires à la redéfinition d'un caractère ou d'un lutin.

L'écran du TI affiche une grille de 16 * 16 carrés, représentant 256 pixels; il est donc possible de définir simultanément quatre caractères graphiques. Pour faire passer les pixels de l'état "off" à l'état "on", il

suffit de ce déplacer dans la grille à l'aide des joysticks. Pour faire passer les points de "on" à "off", il faut appuyer sur le "bouton de tir".

Pour mieux juger de l'effet de votre création, le programme affiche le ou les caractères à leur taille réelle et il est possible de changer la couleur des points "on" et "off".

Vous pouvez obtenir l'affichage des codes hexadécimaux à l'écran ou sur

imprimante. Pour le dernier cas, vous obtenez la représentation du caractère sur le listing; la transmission à l'imprimante ne se fait pas en haute résolution, ce qui autorise une adaptation aisée à différents types d'imprimante. Le programme est prévu pour une imprimante parallèle; pour une imprimante série, il faut modifier la ligne 800 (OPEN#1:"RS232.BA=9600", par exemple).

□

```

10 CALL CLEAR :: CALL DELSPRITE(#1):: RE
STORE :: DIM BI(4),HD$(15),CI$(16):: BI(
1)=8 :: BI(2)=4 :: BI(3)=2 :: BI(4)=1
20 FOR B=0 TO 9 :: HD$(B)=STR$(B):: NEXT
B :: FOR B=10 TO 15 :: HD$(B)=CHR$(B+55
):: NEXT B
30 DATA TRANSPA,NOIR,VERT M,VERT C,BLEU
F,BLEU C,ROUGE F,CYAN,ROUGE M,ROUGE C,JA
UNE F,JAUNE C,VERT F,MAGENTA,GRIS,BLANC
100 A$="FF80808080808080" :: B$=RPT$("F",
16):: CALL CHAR(96,A$,97,RPT$("80",8),98
,"FF",99,B$,104,A$,112,B$):: CALL COLOR(
10,2,9)
110 CALL HCHAR(17,3,98,16):: FOR B=1 TO
16 :: CALL HCHAR(B,3,96,16):: NEXT B ::
CALL VCHAR(1,19,97,16):: CALL HCHAR(8,8,
104):: PH,PV=8
120 CALL JOYST(1,X,Y):: CALL KEY(1,K,S):
: IF K=18 THEN C=0 :: CALL COLOR(10,2,11
)ELSE C=1 :: CALL COLOR(10,2,9)
130 IF K=19 THEN CALL HCHAR(PV,PH,99)::
GOTO 500
140 IF X=0 AND Y=4 THEN 200 ELSE IF X=4
AND Y=0 THEN 250 ELSE IF X=0 AND Y=-4 TH
EN 300 ELSE IF X=-4 AND Y=0 THEN 350·ELS
E 120
200 IF PV=1 THEN 120 ELSE PV=PV-1 :: GOS
UB 2000
210 CALL HCHAR(PV+1,PH,CA):: GOTO 400
250 IF PH=18 THEN 120 ELSE PH=PH+1 :: GO
SUB 2000
260 CALL HCHAR(PV,PH-1,CA):: GOTO 400
300 IF PV=16 THEN 120 ELSE PV=PV+1 :: GO
SUB 2000
310 CALL HCHAR(PV-1,PH,CA):: GOTO 400
350 IF PH=3 THEN 120 ELSE PH=PH-1 :: GOS
UB 2000
360 CALL HCHAR(PV,PH+1,CA)
400 FOR B=0 TO 20 :: NEXT B :: GOTO 120
500 FOR B1=1 TO 8 :: FOR B2=3 TO 6 :: CA
LL GCHAR(B1,B2,COC):: IF COC=99 THEN COM
P=COMP+BI(B2-2)
510 NEXT B2 :: GOSUB 2100
520 FOR B3=7 TO 10 :: CALL GCHAR(B1,B3,C
OC):: IF COC=99 THEN COMP=COMP+BI(B3-6)
525 NEXT B3 :: GOSUB 2100 :: NEXT B1
530 FOR B4=9 TO 16 :: FOR B5=3 TO 6 :: C
ALL GCHAR(B4,B5,COC):: IF COC=99 THEN CO
MP=COMP+BI(B5-2)
540 NEXT B5 :: GOSUB 2100
550 FOR B6=7 TO 10 :: CALL GCHAR(B4,B6,C
OC):: IF COC=99 THEN COMP=COMP+BI(B6-6)
560 NEXT B6 :: GOSUB 2100 :: NEXT B4
570 FOR B7=1 TO 8 :: FOR B8=11 TO 14 ::
CALL GCHAR(B7,B8,COC):: IF COC=99 THEN C
OMP=COMP+BI(B8-10)
575 NEXT B8 :: GOSUB 2100
580 FOR B9=15 TO 18 :: CALL GCHAR(B7,B9,
COC):: IF COC=99 THEN COMP=COMP+BI(B9-14
)
590 NEXT B9 :: GOSUB 2100 :: NEXT B7
600 FOR B10=9 TO 16 :: FOR B11=11 TO 14
:: CALL GCHAR(B10,B11,COC):: IF COC=99 T
HEN COMP=COMP+BI(B11-10)
610 NEXT B11 :: GOSUB 2100
620 FOR B12=15 TO 18 :: CALL GCHAR(B10,B
12,COC):: IF COC=99 THEN COMP=COMP+BI(B1
2-14)
630 NEXT B12 :: GOSUB 2100 :: NEXT B10
700 DISPLAY AT(17,1):"CODE HEXADECIMAL:"
:: DISPLAY AT(18,1):CHaine$
710 RESTORE :: CALL DELSPRITE(#1):: FOR
B=1 TO 16 :: DISPLAY AT(B,18):USING "##"
:B :: READ COU$ :: DISPLAY AT(B,21):COU$

```

```

:: NEXT B
720 DISPLAY AT(22,1):"" :: DISPLAY AT(23,1):"COULEUR DU DESSIN?" :: ACCEPT AT(23,20)VALIDATE(DIGIT):CODE :: IF CODE<1 OR CODE>16 THEN 720
730 DISPLAY AT(24,1):"COULEUR DE FOND?" :: ACCEPT AT(24,18)VALIDATE(DIGIT):COFO :: IF COFO<1 OR COFO>16 THEN 730
740 FOR B=1 TO 16 :: DISPLAY AT(B,18):"" :: NEXT B :: CALL COLOR(11,COFO,1)
750 FOR B=1 TO 6 :: CALL HCHAR(B,22,112,6):: NEXT B :: CALL CHAR(120,CHaine$):: CALL MAGNIFY(3)
760 CALL SPRITE(#1,120,CODE,17,185)
770 DISPLAY AT(21,1):"1-MODIFICATION COULEURS" :: DISPLAY AT(22,1):"2-MODIFICATION DESSIN" :: DISPLAY AT(23,1):"3-NOUVEAU DESSIN"
775 DISPLAY AT(24,1):"4-IMPRESSION 5-FIN"
780 CALL KEY(1,K1,S1):: IF S1=0 THEN 780
785 FOR B=21 TO 24 :: DISPLAY AT(B,1):"" :: NEXT B

```

```

786 IF K1=9 THEN 800 ELSE IF K1=10 THEN CALL DELSPRITE(#1):: CALL CLEAR :: END
790 IF K1=19 THEN 710 ELSE IF K1=7 THEN CHaine$="" :: CALL HCHAR(PV,PH,104):: GO TO 120 ELSE IF K1=8 THEN CHaine$="" :: GOTO 10 ELSE 770
800 OPEN #1:"P10.DF"
810 FOR BL=1 TO 16 :: FOR BC=3 TO 18 :: CALL GCHAR(BL,BC,C1)
820 IF C1=99 THEN CI$(BL)=CI$(BL)&"#" ELSE CI$(BL)=CI$(BL)&" "
830 NEXT BC :: NEXT BL
840 FOR BIM=1 TO 16 :: PRINT #1:CHR$(14);TAB(5);CI$(BIM):: NEXT BIM
870 PRINT #1:CHR$(15);"":"CODE HEXADECIMAL":CHaine$
880 CLOSE #1 :: FOR BEF=1 TO 16 :: CI$(BEF)="" :: NEXT BEF :: GOTO 770
2000 CALL HCHAR(PV,PH,104):: IF C THEN CA=99 ELSE CA=96
2010 RETURN
2100 CHaine$=CHaine$&HD$(COMP):: COMP=0 :: RETURN

```

Kaleidoscope

Marianne Sutz

Ce petit programme en **Basic Etendu** se passe de commentaires, et prouve, si besoin était, que le TI est un très bon outil graphique puisqu'il permet d'obtenir des effets extrêmement intéressants avec seulement quelques lignes de Basic.

□

```

100 CALL SCREEN(1):: CALL CLEAR :: RANDO MIZE :: OPTION BASE 1 :: DIM COO(14),COF(14)
110 FOR B=33 TO 137 STEP 8 :: CALL CHAR(B,"0F0F0F0F0F0F0F0F",B+1,"F0F0F0F0F0F0F0F0",B+2,"0F0F0F0F0F0F0F0F"):: NEXT B
120 FOR B=1 TO 14 :: CALL COLOR(B,16,1):: COF(B),COO(B)=1 :: NEXT B
150 FOR L=1 TO 6 :: FOR C=1 TO 8 :: CALL HCHAR(L,C,(25+(INT(RND*14)+1)*8)+INT(RND*2)):: NEXT C :: NEXT L
160 FOR L=1 TO 6 :: FOR C=1 TO 8 :: CALL

```

```

GCHAR(L,C,X):: CALL HCHAR(L,17-C,X+1):: CALL HCHAR(L,16+C,X):: CALL HCHAR(L,33-C,X+1)
161 CALL HCHAR(13-L,C,X+1):: CALL HCHAR(13-L,17-C,X):: CALL HCHAR(13-L,16+C,X+1):: CALL HCHAR(13-L,33-C,X):: CALL HCHAR(12+L,C,X)
162 CALL HCHAR(12+L,17-C,X+1):: CALL HCHAR(12+L,16+C,X):: CALL HCHAR(12+L,33-C,X+1):: CALL HCHAR(25-L,C,X+1)
163 CALL HCHAR(25-L,17-C,X):: CALL HCHAR(25-L,16+C,X+1):: CALL HCHAR(25-L,33-C,X):: NEXT C :: NEXT L
170 CALL KEY(0,K,S):: IF S THEN CALL CLEAR :: GOTO 150 ELSE CAR=INT(RND*14)+1 :: COO(CAR)=INT(RND*16)+1 :: CALL COLOR(CAR,COO(CAR),COF(CAR))
180 CALL KEY(0,K,S):: IF S THEN CALL CLEAR :: GOTO 150 ELSE CAR=INT(RND*14)+1 :: COF(CAR)=INT(RND*16)+1 :: CALL COLOR(CAR,COO(CAR),COF(CAR))
185 GOTO 170

```

Opérations

Albert Rigol

Ce programme éducatif, écrit en Basic TI, permet de faire réciter les tables des quatre opérations à de jeunes écoliers.

Il propose :

- le choix de l'opération;
- le choix de la table;
- une série de dix questions dans un ordre aléatoire;
- si deux erreurs à une même question : affichage de la bonne réponse;
- trois niveaux de notation avec appréciations à chaque niveau;

- selon la note attribuée, l'affichage automatique de la table pour faciliter la révision.

Mode d'emploi

Après le générique d'introduction, le programme commence par demander le prénom de l'enfant, ce qui permettra de personnaliser le programme. Ensuite, le menu s'affiche, on choisit l'opération (multiplication, soustraction, addition et division) puis la table.

Alors, une série de dix questions est posée, ce qui permet de faire réciter toute la table dans un ordre aléatoire. On entre la réponse, puis <ENTER>.

A la fin de la série, la note s'affiche, ainsi qu'une appréciation. Une note égale ou inférieure à 6/10 provoque l'affichage de la table en question, pour la révision tout le temps désiré. On répond ensuite à la question "Une autre série ? O/N".

□

```

100 REM*****
110 REM*   OPERATIONS   *
120 REM*   ALBERT RIGAL *
130 REM*****02.1984*****
140 REM*   POUR TI 99/4A *
150 REM*   TI BASIC    *

160 REM*****
170 GOSUB 3070
180 CALL CLEAR
190 RANDOMIZE
200 CALL SCREEN(7)
210 FOR I=5 TO 8
220 CALL COLOR(I,4,1)
230 NEXT I
240 FOR I=1 TO 4
250 CALL COLOR(I,16,1)
260 NEXT I
270 CALL CLEAR
280 INPUT "QUEL EST TON PRENOM

      S'IL TE PLAIT
? ";P$
290 CALL CLEAR
300 PRINT "MAINTENANT ";P$;"

      AU TRAVAIL!!
..": : : : : : : : : :
310 CALL SOUND(100,800,0)
320 FOR I=1 TO 600
330 NEXT I
340 REM ROUTINE TRANSPARENCE

350 CALL CLEAR
360 FOR I=1 TO 8
370 CALL COLOR(I,1,1)
380 NEXT I
390 REM MENU
400 PRINT "QUELLE OPERATION CH
OISIS-TU?": :
410 PRINT ".....
.....": : : :

```

```

420 PRINT "   MULTIPLICATION
:1": :
430 PRINT "   SOUSTRACTION
:2": :
440 PRINT "   ADDITION
:3": :
450 PRINT "   DIVISION
:4": : : :
460 PRINT ".....TAPE
UN CHIFFRE DE 1 A 4"
470 REM COULEURS CARACTERES
480 FOR I=5 TO 8
490 CALL COLOR(I,4,1)
500 NEXT I
510 FOR I=1 TO 4
520 CALL COLOR(I,16,1)
530 NEXT I
540 CALL KEY(0,K,S)
550 IF (S=0)+(K<49)+(K>52)THEN
540
560 K=K-48
570 ON K GOSUB 590,1200,1810,2
420
580 GOTO 350
590 CALL CLEAR
600 REM MULTIPLICATION
610 PRINT "QUELLE TABLE

      DE MULTIPLICA
TION

      CHOISIS-TU ? DE 1
A 9": : : :
620 PRINT "   tape u
n chiffre"
630 CALL KEY(5,K,S)
640 IF (S=0)+(K<49)+(K>57)THEN
630
650 A=K-48
660 CALL CLEAR
670 X=0
680 DIM N(10)
690 FOR I=1 TO 10
700 N(I)=I
710 NEXT I
720 FOR I=10 TO 1 STEP -1
730 R=INT(RND*1)+1

```

```

740 Q=A*N(R)
750 INPUT STR$(A)&" FOIS "&STR
$(N(R))&" EGAL? ";T
760 IF T<>Q THEN 800
770 PRINT ":"
JUSTE...": :
780 Z=0
790 GOTO 880
800 Z=Z+1
810 IF Z=2 THEN 3800
820 PRINT ":" ERREUR! ";P$
: : " RECOMMENCE
...": :
830 X=X+1
840 CALL SOUND(500,200,0)
850 FOR V=1 TO 300
860 NEXT V
870 GOTO 750
880 IF R=I THEN 920
890 FOR J=R+1 TO I
900 N(J-1)=N(J)
910 NEXT J
920 NEXT I
930 Y=10-X
940 IF Y=9 THEN 970
950 IF (Y=8)+(Y=7)THEN 1030
960 IF Y<=6 THEN 1070
970 PRINT "-----
-----": "TRES BIEN ";P$;
"
BRA
VO!!!": :
980 CALL SOUND(200,400,1)
990 CALL SOUND(200,500,1)
1000 CALL SOUND(200,600,1)
1010 CALL SOUND(200,800,1)
1020 GOTO 1090
1030 PRINT "-----
-----": "BIEN !. ";P$;"
": :
1040 CALL SOUND(200,600,1)
1050 CALL SOUND(200,800,1)
1060 GOTO 1090
1070 PRINT "-----
-----": "PAS FAMEUX !.
";P$;"
FAUDRAS REVISER

```

```

LA TABLE...": : :
1080 CALL SOUND(200,800,1)
1090 PRINT : "RESULTATS :";Y;"SU
R 10...": :-----
": :
1100 FOR E=1 TO 800
1110 NEXT E
1120 IF Y>=7 THEN 1140
1130 GOSUB 3680
1140 PRINT : "UNE AUTRE SERIE ?
O/N": :
1150 CALL KEY(0,K,S)
1160 IF (S=0)+(K<78)+(K>79)THE
N 1150
1170 IF K=78 THEN 3020
1180 RETURN
1190 REM SOUSTRACTION
1200 CALL CLEAR
1210 CALL SCREEN(14)
1220 PRINT : "QUELLE TABLE DE S
OUSTRACTION
CHOISIS-TU
? DE 1 A 9": : : :
1230 PRINT " tape
un chiffre"
1240 CALL KEY(5,K,S)
1250 IF (S=0)+(K<49)+(K>57)THE
N 1240
1260 A=K-48
1270 CALL CLEAR
1280 X=0
1290 FOR I=1 TO 10
1300 N(I)=I
1310 NEXT I
1320 FOR I=10 TO 1 STEP -1
1330 Z=0
1340 R=INT(RND*I)+1
1350 INPUT STR$(N(R)+10)&" MOI
NS "&STR$(A)&" EGAL? ":T
1360 Q=N(R)+10-A
1370 IF T<>Q THEN 1410
1380 PRINT : "
VRAI..."
1390 Z=0
1400 GOTO 1490
1410 Z=Z+1
1420 IF Z=2 THEN 3950
1430 PRINT : " FAUX ! ";P
$;: : " RECOMMENC
E...": :
1440 X=X+1
1450 CALL SOUND(500,200,0)
1460 FOR V=1 TO 300
1470 NEXT V
1480 GOTO 1350
1490 IF R=1 THEN 1530
1500 FOR J=R+1 TO I
1510 N(J-1)=N(J)
1520 NEXT J
1530 NEXT I
1540 Y=10-X
1550 IF Y>=9 THEN 1580
1560 IF (Y=8)+(Y=7)THEN 1640
1570 IF Y<=6 THEN 1680
1580 PRINT : "
": : "BRAVO!! ";P$;
TRES BIEN...": :
1590 CALL SOUND(200,400,1)
1600 CALL SOUND(200,500,1)
1610 CALL SOUND(200,600,1)
1620 CALL SOUND(200,800,1)
1630 GOTO 1700
1640 PRINT : "
": : "BIEN!!...";P$;:
1650 CALL SOUND(200,600,1)
1660 CALL SOUND(200,800,1)

```

```

1670 GOTO 1700
1680 PRINT : "
": : "MEDIOCRE !!. "
P$;
ETUDIE LA TABLE.
": :
1690 CALL SOUND(200,800,1)
1700 PRINT : "RESULTATS :";Y;"S
UR 10...": :-----
": :
1710 FOR E=1 TO 800
1720 NEXT E
1730 IF Y>=7 THEN 1750
1740 GOSUB 3830
1750 PRINT : "UNE AUTRE SERIE?
O/N"
1760 CALL KEY(5,K,S)
1770 IF (S=0)+(K<78)+(K>79)THE
N 1760
1780 IF K=78 THEN 3020
1790 RETURN
1800 REM *ADDITION*****
1810 CALL CLEAR
1820 CALL SCREEN(6)
1830 PRINT : "QUELLE TABLE D'AD
DITION
CHOISIS-TU
? DE 1 A 9": : : :
1840 PRINT " tape
un chiffre"
1850 CALL KEY(5,K,S)
1860 IF (S=0)+(K<49)+(K>57)THE
N 1850
1870 A=K-48
1880 CALL CLEAR
1890 X=0
1900 FOR I=1 TO 10
1910 N(I)=I
1920 NEXT I
1930 FOR I=10 TO 1 STEP -1
1940 Z=0
1950 R=INT(RND*I)+1
1960 Q=A+N(R)
1970 INPUT STR$(A)&" PLUS "&ST
R$(N(R))&" EGAL? ":T
1980 IF T<>Q THEN 2020
1990 PRINT : "
EXACT..."
2000 Z=0
2010 GOTO 2110
2020 Z=Z+1
2030 IF Z=2 THEN 4100
2040 PRINT : " NON! ";
P$;: : " RECOMMEN
CE...": :
2050 X=X+1
2060 CALL SOUND(500,200,0)
2070 FOR V=1 TO 300
2080 NEXT V
2090 GOTO 1970
2100 IF R=1 THEN 2140
2110 FOR J=R+1 TO I
2120 N(J-1)=N(J)
2130 NEXT J
2140 NEXT I
2150 Y=10-X
2160 IF Y>=9 THEN 2190
2170 IF (Y=8)+(Y=7)THEN 2250
2180 IF Y<=6 THEN 2290
2190 PRINT : "
": : "C'EST PARFAIT
";P$;
AVO !!!...": :
2200 CALL SOUND(200,400,1)
2210 CALL SOUND(200,500,1)
2220 CALL SOUND(200,600,1)

```

```

2230 CALL SOUND(200,800,1)
2240 GOTO 2310
2250 PRINT : "
": : "BIEN...";P$;:
ETUDIE...": :
2260 CALL SOUND(200,600,1)
2270 CALL SOUND(200,800,1)
2280 GOTO 2310
2290 PRINT : "
": : "PAS BON DU TOU
T !!!...";P$;:
ETUDIE...": :
2300 CALL SOUND(200,800,1)
2310 PRINT : "RESULTATS :";Y;"S
UR 10...": :-----
": :
2320 FOR E=1 TO 800
2330 NEXT E
2340 IF Y>=7 THEN 2360
2350 GOSUB 3980
2360 PRINT : "UNE AUTRE SERIE ?
O/N": :
2370 CALL KEY(5,K,S)
2380 IF (S=0)+(K<78)+(K>79)THE
N 2370
2390 IF K=78 THEN 3020
2400 RETURN
2410 REM DIVISION
2420 CALL CLEAR
2430 CALL SCREEN(7)
2440 PRINT : "QUELLE TABLE DE D
IVISION
CHOISIS-TU
? DE 1 A 9": : : :
2450 PRINT " tape
un chiffre"
2460 CALL KEY(5,K,S)
2470 IF (S=0)+(K<49)+(K>77)THE
N 2460
2480 CALL CLEAR
2490 A=K-48
2500 X=0
2510 FOR I=1 TO 10
2520 N(I)=I
2530 NEXT I
2540 FOR I=10 TO 1 STEP -1
2550 Z=0
2560 R=INT(RND*I)+1
2570 Q=N(R)*A/A
2580 INPUT STR$(N(R)*A)&" QUE
DIVISE "&STR$(A)&" EGAL? ":T
2590 IF T<>Q THEN 2630
2600 PRINT : "
BON...": :
2610 Z=0
2620 GOTO 2710
2630 Z=Z+1
2640 IF Z=2 THEN 4250
2650 PRINT : " FAUX !. ";P$;
: : " RECOMMENCE.
...": :
2660 X=X+1
2670 CALL SOUND(500,200,0)
2680 FOR V=1 TO 300
2690 NEXT V
2700 GOTO 2580
2710 IF R=1 THEN 2750
2720 FOR J=R+1 TO I
2730 N(J-1)=N(J)
2740 NEXT J
2750 NEXT I
2760 Y=10-X
2770 IF Y>=9 THEN 2800
2780 IF (Y=8)+(Y=7)THEN 2860
2790 IF Y<=6 THEN 2900
2800 PRINT : "
": : "TRES BIEN ";P$

```



```

; " BRAVO !!!" :
2810 CALL SOUND(200,400,1)
2820 CALL SOUND(200,500,1)
2830 CALL SOUND(200,600,1)
2840 CALL SOUND(200,800,1)
2850 GOTO 2920
2860 PRINT : "-----"
;P$;"":
2870 CALL SOUND(200,600,1)
2880 CALL SOUND(200,800,1)
2890 GOTO 2920
2900 PRINT : "-----"
; "P$;"
      TU NE SAIS PAS
LA TABLE..." :
2910 CALL SOUND(200,800,1)
2920 PRINT : "RESULTATS :";Y;"S
UR 10..." : "-----"
2930 FOR E=1 TO 800
2940 NEXT E
2950 IF Y>=7 THEN 2970
2960 GOSUB 4130
2970 PRINT : "UNE AUTRE SERIE ?
O/N"
2980 CALL KEY(5,K,S)
2990 IF (S=0)+(K<78)+(K>79)THE
N 2980
3000 IF K=78 THEN 3020
3010 RETURN
3020 CALL CLEAR
3030 GOSUB 4290
3040 CALL CLEAR
3050 END
3060 REM AFFICHAGE GENERIQUE
3070 CALL CLEAR
3080 CALL SCREEN(16)
3090 CALL CHAR(90,"FFFFFFFFFFFF
FFFFF")
3100 CALL CHAR(130,"FFFFFFFFFFFF
FFFFF")
3110 CALL CHAR(136,"FFE7C38181
C3E7FF")
3120 FOR T=1 TO 14
3130 CALL COLOR(T,16,1)
3140 NEXT T
3150 CALL HCHAR(4,9,90,19)
3160 CALL HCHAR(5,10,136,17)
3170 CALL VCHAR(6,10,136,13)
3180 CALL VCHAR(5,9,90,15)
3190 CALL VCHAR(5,27,90,15)
3200 CALL VCHAR(6,26,136,13)
3210 CALL HCHAR(6,11,130,15)
3220 CALL HCHAR(7,11,130,15)
3230 CALL HCHAR(8,11,130,15)
3240 CALL HCHAR(9,11,130,15)
3250 CALL HCHAR(10,11,130,15)
3260 CALL HCHAR(11,11,130,15)
3270 CALL HCHAR(12,11,136,15)
3280 CALL HCHAR(13,11,130,15)
3290 CALL HCHAR(14,11,130,15)
3300 CALL HCHAR(15,11,130,15)
3310 CALL HCHAR(16,11,130,15)
3320 CALL HCHAR(17,11,130,15)
3330 CALL HCHAR(18,11,130,15)
3340 CALL HCHAR(19,10,136,17)
3350 CALL HCHAR(20,9,90,19)
3360 A$="PROGRAMME"
3370 B$="-OPERATIONS-"
3380 C$=" * / + - "
3390 E$="02.1984"
3400 FOR I=1 TO LEN(A$)
3410 CALL HCHAR(7,13+I,ASC(SEG
$(A$,I,1)))

```

```

3420 NEXT I
3430 FOR I=1 TO LEN(B$)
3440 CALL HCHAR(10,11+I,ASC(SE
G$(B$,I,1)))
3450 NEXT I
3460 FOR I=1 TO LEN(C$)
3470 CALL HCHAR(14,13+I,ASC(SE
G$(C$,I,1)))
3480 NEXT I
3490 FOR I=1 TO LEN(E$)
3500 CALL HCHAR(17,14+I,ASC(SE
G$(E$,I,1)))
3510 NEXT I
3520 CALL SCREEN(2)
3530 CALL COLOR(8,5,1)
3540 FOR R=1 TO 3
3550 C=1
3560 FOR I=1 TO 16
3570 A=INT(RND*16)+1
3580 CALL COLOR(13,C,1)
3590 CALL COLOR(14,3,A)
3600 C=C+1
3610 A=A+1
3620 FOR P=1 TO 8
3630 NEXT P
3640 NEXT I
3650 NEXT R
3660 RETURN
3670 REM AFFICHAGE DE LA TABLE
ET BONNE REPONSE
3680 CALL CLEAR
3690 FOR B=1 TO 10
3700 PRINT A;"X";B;"=";A*B:
3710 NEXT B
3720 A$="COURAGE..."
3730 FOR I=1 TO LEN(A$)
3740 CALL HCHAR(12,16+I,ASC(SE
G$(A$,I,1)))
3750 NEXT I
3760 FOR I=1 TO LEN(P$)
3770 CALL HCHAR(14,21+I,ASC(SE
G$(P$,I,1)))
3780 NEXT I
3790 RETURN
3800 PRINT : "ERREUR !!!";P$
: " " "&STR$(A)&" fois
"&STR$(N(R))&" egal
";A*N(R):
3810 Z=0
3820 GOTO 880
3830 CALL CLEAR
3840 FOR B=11 TO 20
3850 PRINT B;"-";A;"=";B-A:
3860 NEXT B
3870 A$="ENCORE UN EFFORT."
3880 FOR I=1 TO LEN(A$)
3890 CALL HCHAR(12,14+I,ASC(SE
G$(A$,I,1)))
3900 NEXT I
3910 FOR I=1 TO LEN(P$)
3920 CALL HCHAR(14,18+I,ASC(SE
G$(P$,I,1)))
3930 NEXT I
3940 RETURN
3950 PRINT : "FAUX!!...";P$
: " " "&STR$(N(R)+10)&"
moins "&STR$(A)&" eg
al";N(R)+10-A:
3960 Z=0
3970 GOTO 1490
3980 CALL CLEAR
3990 FOR B=1 TO 10
4000 PRINT A;"+";B;"=";A+B:
4010 NEXT B
4020 A$=" REVISE!!!"
4030 FOR I=1 TO LEN(P$)

```

```

4040 CALL HCHAR(12,16+I,ASC(SE
G$(P$,I,1)))
4050 NEXT I
4060 FOR I=1 TO LEN(A$)
4070 CALL HCHAR(14,20+I,ASC(SE
G$(A$,I,1)))
4080 NEXT I
4090 RETURN
4100 PRINT : "NON !!...";P$
: " " "&STR$(A)&" plus "
&STR$(N(R))&" egal";
A+N(R):
4110 Z=0
4120 GOTO 2100
4130 CALL CLEAR
4140 FOR B=1 TO 10
4150 PRINT B*A;" / ";A;"=";B*A/A
:
4160 NEXT B
4170 A$="ETUDIE!!!"
4180 FOR I=1 TO LEN(A$)
4190 CALL HCHAR(12,18+I,ASC(SE
G$(A$,I,1)))
4200 NEXT I
4210 FOR I=1 TO LEN(P$)
4220 CALL HCHAR(14,20+I,ASC(SE
G$(P$,I,1)))
4230 NEXT I
4240 RETURN
4250 PRINT : " NON !!...!"
;P$;: " " "&STR$(N(R)*A)&"
que divise "&STR$(A)
&" egal";N(R)*A/A:
4260 Z=0
4270 GOTO 2710
4280 REM FIN DU PROGRAMME
4290 CALL CLEAR
4300 CALL CHAR(128,"3078793276
FCFCFC")
4310 CALL CHAR(129,"307878307F
FCFCFC")
4320 CALL CHAR(130,"FCFCFC4848
484848")
4330 CALL COLOR(13,16,1)
4340 PRINT " * AU REVOI
R *":
4350 FOR I=1 TO 15
4360 CALL SOUND(5,1600,2)
4370 CALL HCHAR(18,13,128,9)
4380 CALL HCHAR(19,13,130,9)
4390 FOR J=1 TO 50
4400 NEXT J
4410 CALL HCHAR(18,13,129,9)
4420 NEXT I
4430 RETURN

```

Les cassettes
" 99 Magazine "
1 à 8 contiennent tous
les programmes publiés
dans les revues correspon-
dantes, à l'exception des
sources Assembleur ou
Pascal...

Un "bug" dans "GALAXIA"

Une erreur apparaissait sur le listing publié dans le précédent numéro de "99 Magazine", ainsi que sur le code source placé sur la disquette correspondante. Toutefois, le code objet, baptisé "GALAXIA", était correct. Pour que tout rentre dans l'ordre, il faut modifier, dans le fichier source "GALAXIA-0", la ligne placée juste avant l'étiquette KEY :

BL @VSBW au lieu de BLWP @VSBW

dans le fichier "GALAXIA-6", il faut supprimer la dernière valeur de DATA (.288) dans l'avant dernière ligne.

Quelques adresses

Jean-Claude Hardy, un de nos fidèles lecteurs de Montréal, nous transmet quelques adresses de distributeurs de logiciels pour le TI-99/4A.

- TEX_COMP, P.O. BOX 33084, Granada Hill, Ca. 91344, U.S.A. Tél. : (818) 366-6631.
- TRITON, Products Company, Order Center, P.O. BOX 8123, San Francisco, Ca, 94128. Tél. : (800) 227-6900.
- Compucentre, 9001 Boul. Louis-H. Lafontaine, Montréal, Canada, Tél. : (514) 354-3810.

Petites annonces (gratuites)

• A vendre, TI-99/4A T.B.E. 1000 FF. Module "Basic Etendu" + manuel 500 FF. Extension de mémoire externe 32Ko (01/85) 1100 FF. Magnéto Philips D6600 + câble 550 FF. Paire de joysticks 150 FF. Nombreux livres et cassettes (jeux, utilitaires...). Le tout pour 3000 FF (avec livres et cassettes). Bertrand Delestrez - 4, rue Grissom - 62210 AVION - (21) 28.21.75.

• Vends modules "Parsec", "Burger Time", "Startrek", "Fathom", "Microsurgeon" : 180 FF. "Tombstone", "Video 1", "Wumpus" : 100 FF. Cassettes "Lunar Lander", "Jumper", "Solar System" : 80 FF. Livres. J. Reibel - 9, square Fleming - 92350 LE PLESSIS-ROBINSON - (1) 631.46.11.

• Cherche (échange) programmes pour TI (Basic TI ou Basic Etendu). Réponse assurée. Cherche aussi un manche de manette de jeu. Laurent Kraft - 22, rue Pasteur - 68300 SAINT-LOUIS - (89) 67.51.18.

• Vends TI-99, interface SECAM, manettes de jeux, cordon magnéto, Basic Etendu, Mini-mémoire, manuels, cassettes "Basic par soi-même", "Conseils financiers", "Sungames" et 50 programmes sur cassettes. 10 Hebdogiciels. 5 modules de jeux ("Pole position", "Moon Patrol", "Foot", "Carwars", "TI Invaders"). Valeur de l'ensemble 4800 FF, cédé à 3500 FF. Xavier JACQ - 8, rue de la gare - 29223 SAINT THEGONNEC - (98) 79.42.43.

• Achète "Editeur/Assembleur", "Synthétiseur de paroles" et "Terminal Emulator II" pour TI 99/4A. Claude Besneux - BP 56 - 92110 CLICHY - 564.17.74 de 14h à 17h.

• Recherche cassette "Assembleur ligne par ligne" (ou copie) utilisable avec la "Mini-mémoire" (écrasée involontairement et irrécupérable !). Dédommagement assuré. (6) 448.61.81 après 20h.

• Vends module "Gestion de fichiers" (250 FF) et cassette "Conseil financier" (50 FF). Jean-François Dupont - 8, rue Madame de Sévigné - 53000 LAVAL.

• Vends TI-99/4A avec 9 modules : "Basic Etendu", "Echecs", "Othello", "Parsec", "MunchMan", "TI Invaders", "Chasse au Wumpus", "Cars War", "Pirat's Adventure" et synthétiseur de paroles, cordon magnéto, 5 manuels, manettes de jeux. Très bon état. Prix à débattre. Patrick - 657.22.66 entre 18 et 19h.

• Vends boîtier d'extension, lecteur de disquettes et carte contrôleur. Mini-mémoire, livre Assembleur. Cassettes Basic et Basic Etendu par soi-même. Cordon magnéto, 4 modules de jeux, cassettes "Lunar Lander II". 3 livres de programmes. Marc Mathonnet - 33, avenue Auguste Blanqui - 69100 VILLEURBANNE - 16 (7) 885.73.78.

• Vends TI-99/4A, manettes de jeux, magnétophone à cassettes. Guy Forestier - 33910 TAURIAC - (57) 68.86.05 après 18h.

• Vends "Mini-mémoire" et livre "Initiation à l'assembleur" : 750 FF. module "Foot" : 250 FF, "Music-Maker" : 250 FF. François Frattini - 30, galerie de l'Arlequin - Apt. 2224 - 38100 GRENOBLE - (76) 22.16.12.

• Vends TI-99/4A, "Basic Etendu" en français, livres et cassettes : 2000 FF. (70) 20.13.84 ou (37) 97.21.72 le week end.

• Cherche modules "Mini-mémoire" avec cassette et "Basic Etendu". Paul Ducros - 55, rue des Lorlots - 03100 MONTLUÇON.

• Cherche contacts sur SAUMUR et sa région pour tester programme. (41) 38.11.90 le matin.

• Vends "Mini-mémoire" avec manuel "Editeur/Assembleur" et "Basic Etendu". (1) 704.43.17.

• Achète module "Tunnels of Doom" + 2 cassettes (environ 250 FF). Achète ou échange cassettes "Mystery Fun House", "Mission Impossible", "Voodoo Castle" (je peux échanger contre d'autres jeux d'Adventure International). Laurent Pelé - 5, clos d'Ergal - 78310 ELANCOURT - (3) 051.18.96.

99 Magazine s'adresse...

... aux débutants

Vous venez d'acquérir un TI-99/4A (on en trouve encore !), Tout est maintenant déballé et installé et vous êtes en train de découvrir qu'un ordinateur, ça ne marche pas tout seul... Il existe bien des brochures d'accompagnement, mais elles ne sont pas toujours rédigées de façon claire. En outre, le seul renseignement dont vous avez besoin en ce moment ne s'y trouve peut-être pas.

"99" vous apporte chaque trimestre de nombreux renseignements, des programmes complets, des exemples. De plus, "99" édite avec chaque numéro une cassette regroupant tous les programmes du numéro. Vous pouvez ainsi utiliser des programmes que vous y avez vus, même si vous n'avez pas complètement compris le listing. En effet, vous savez bien utiliser des jeux ou des programmes de gestion de fichiers sans pour autant être capables de comprendre intégralement le listing de ces programmes.

... aux habitués

En étudiant les nombreux programmes que nous vous

proposons, vous pourrez perfectionner votre connaissance de Basic simple et du Basic Etendu mais aussi vous initier au Pascal et à l'assembleur. Dans pratiquement chaque numéro de "99", vous trouverez des recettes ou des programmes pour mieux programmer et utiliser votre TI-99.

... aux experts

Vous êtes maintenant nombreux à mériter le titre d'experts. Chacun d'entre vous connaît plus particulièrement un langage ou un domaine d'application. "99", en publiant les écrits d'autres experts, peut vous en apprendre plus, que ce soit dans votre domaine de prédilection ou dans d'autres domaines.

... à tous ses lecteurs

Quel que soit votre niveau, vous pouvez avoir des informations, des idées ou des programmes à transmettre aux autres. Faites nous les parvenir afin que nous puissions faire partager ce savoir que vous avez acquis. L'idéal est de nous envoyer une cassette ou disquette et un article dactylographié.

99 MAGAZINE

BULLETIN D'ABONNEMENT

Je désire recevoir pour le N° ☐ de 99 magazine
☐ le numéro avec cassette 95 F TTC
☐ le numéro sans cassette 40 F TTC
☐ la cassette seule 55 F TTC

Je désire m'abonner pour 4 numéros
à partir du N° ☐
☐ avec cassette 325 F TTC
☐ sans cassette 135 F TTC

Nom
Adresse

Ces tarifs comprennent l'envoi postal en France Métropolitaine et CEE (voie aérienne exceptée)
Envoi par avion : nous consulter

Envoyez ce bon de commande et votre règlement à :

Editions MEV - 49 rue Lamartine - 78000 Versailles

Je vous écris pour vous féliciter et ensuite vous poser quelques questions :

Comment, avec l'assembleur (mini-mémoire et extension de mémoire) envoyer quelque chose vers l'imprimante ? existe-t-il une adresse ? Je possède personnellement l'imprimante Seikosha GP100A. Comment alors simuler le CHR\$(8) (accès au mode graphique) ?

Ma seconde question porte sur la routine DSRLNK (à l'adresse >6038 pour la mini-mémoire) : à quoi sert-elle ? Je crois qu'elle n'est même pas mentionnée dans le livre de Denise Amrouche et Roger Didi intitulé "Initiation au langage assembleur de l'ordinateur familial Texas Instruments".

J'aimerais savoir si, en mode 40 colonnes (par l'assembleur), on peut, simplement et sans revenir systématiquement au mode 32 colonnes, mettre en mémoire (ou afficher sur l'écran) une lettre frappée sur le clavier.

Stéphane Dunglas - 121, rue de la Pompe - 75116 Paris.

Votre seconde question constitue une réponse à la première. En effet, la routine DSRLNK est utilisée pour effectuer toutes les sorties (ou entrées) vers (ou depuis) les périphériques. Nous avons prévu un article consacré aux diverses utilisations de cette routine mais, faute de place, il nous était impossible de le publier dans ce numéro; vous le trouverez donc dans le numéro 9. En attendant, des exemples concrets sont à votre disposition dans les programmes "Désassembleur" (numéro 5) et "Routines graphiques" (numéro 7).

Le clavier et le processeur vidéo sont en fait deux "périphériques" totalement indépendants; il est donc possible d'obtenir un affichage 40 colonnes tout en ayant toujours accès au clavier. Pour vous en convaincre, vous pouvez consulter le programme "Désassembleur" (encore !) du numéro 5.

Je vous communique une petite adresse utile :

En Basic Etendu (avec extension de mémoire) ou mini-mémoire, entrer un programme (n'importe quel), puis :

CALL INIT

CALL PEEK(-J1952,A,B,C,D)
PRINT A,B,C,D
(notez les valeurs affichées)
NEW (plus de programme !)

Tapez maintenant :

CALL LOAD(-J1952,A,B,C,D)
(A, B, C et D étant les valeurs trouvées précédemment) et
LIST

Votre programme est toujours là.

Conclusion : le NEW n'efface pas le programme en mémoire (cela marche même si vous éteignez la console, mais pas le boîtier d'extension).

Omar Belmoktar - 6, allée S. de Brosse - 93150 Le Blanc Mesnil.

Est-il possible de connecter une table traçante sur l'interface RS232.

Docteur G. Lourde - 8, place du Commando - 44600 St Nazaire Cidex.

L'interface RS232 du TI-99 est standard et, de plus, configurable par logiciel; une table traçante répondant à ce standard doit donc fonctionner normalement sur le TI. Cependant, si vous projetez l'acquisition d'un tel appareil, nous ne saurions trop vous conseiller de faire un essai chez votre revendeur.

Pourriez-vous m'indiquer les noms des MODEMS pouvant être raccordés au TI-99/4A.

Arnaud Duminy - chemin des Ecus - 59600 Maubeuge.

On peut, par l'intermédiaire de l'interface série, raccorder n'importe quel MODEM au TI. Il existe de nombreux MODEM à des prix et avec des possibilités diverses; le mieux serait donc de consulter votre revendeur qui, en fonction de votre problème, pourrait vous conseiller un modèle plutôt qu'un autre.

Je ne possède qu'un seul lecteur de disquette. Quelle est la marche à suivre pour copier une disquette sur une autre disquette vierge. Par exemple, un fichier utilisé pour un annuaire téléphonique, ou encore la disquette "Editeur / Assembleur" pour laquelle le manuel conseille une copie avant

Courrier des Lecteurs

Alexandre Duback

toute utilisation.

Jean Langlet - 18, rue E. Brasselet - Leval - 59620 Aulnoye Aymeries.

Le module "Gestion de disques" est prévu pour ce type d'utilisation. Dans le menu principal (après le menu proposant le choix de la langue utilisée pour les messages), il faut choisir l'option "INITIALISATION POUR UTILISATION AVEC UN SEUL DISQUE". Après cela, il vous suffit de choisir l'option "COPIE DE DISQUE"; le système affichera à l'écran toutes les indications nécessaires.

Etant possesseur d'un TI-99, j'aimerais savoir si un lecteur de disquettes de marque QUMETRACH ou SHUGGART qui utilisent des disquettes double face et double densité peuvent être installés sur le Texas.

Jean-Claude Siracis - 11, rue de la République - 11180 Ouvrillan.

Si vous disposez du contrôleur de disquettes Texas Instruments, le raccordement de ces lecteurs ne devrait pas présenter de difficulté.

Lecteur assidu de "99 Magazine", j'ai lu dans le numéro 7 qu'il était possible de brancher une imprimante ImageWriter sur un TI-99/4A.

Je dispose de la Mini-mémoire, du Basic Etendu, de l'extension 32Ko et de la carte RS232C.

L'imprimante Apple s'adapte-t-elle directement sur le port série de la carte ? Si oui, comment faut-il programmer la carte pour obtenir un listing de programmes écrits en Basic ou Basic Etendu ?

Georges Chareyron - 4, rue F. Buisson - 07400 Le Teil.

L'imprimante Apple "ImageWriter" est aussi facile à utiliser qu'une Epson ou Seikosha. Un câble standard RS232 suffit à son raccordement. Pour obtenir un listing à partir du Basic :

LIST "RS232.BA=9600" (9600 bauds est la valeur par défaut pour l'ImageWriter).

□

pom's

La revue francophone des utilisateurs de l'Apple

Initiation à l'assembleur (7)

Animations graphiques

Copie grand format

M

M

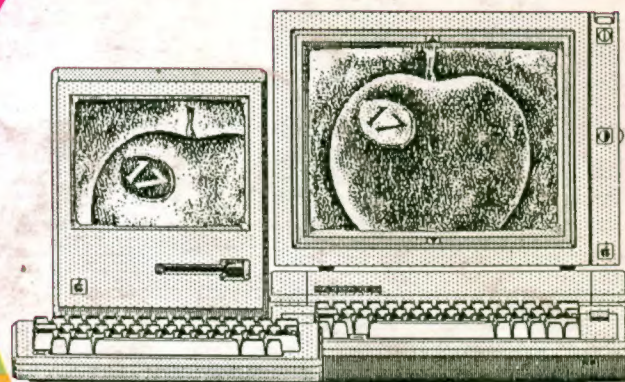
Le

**Pom's, l'indispensable
complément de l'Apple**

des fonctions

ASIC

ateurs en Pascal UCSD



NUMERO 17 • PRIX 40 F

ISSN : 0294-6068

M2366-17-40 F